The statistical properties and coding of handwriting and hand-drawn graphics in tutorial classes

Thesis

Robinson GUIFO GUIFO, B.Sc.(Honours), M.Sc. (U.M.I.S.T)

The Statistical Properties and Coding of Handwriting and Hand-drawn Graphics in Tutorial Classes.

A thesis submitted for the degree of Doctor of philosophy.

Discipline : Electronics

DEDICATION

To my late mother **Elizabeth Moaguo Nkiessu**

AND

To my late father **Antoine TOMAGAM GUIFO GUIFO**

Abstract

This research investigates compression techniques as applied to input from a digitizing tablet. The representations we consider are either drawn from the literature or developed within this thesis; they involve selecting relevant points along the trajectory of the writing pen, and using an interpolating function for reconstruction. The two dimensional parametric polynomial interpolator (e.g linear, quadratic, cubic) is used for regenerating the trajectory of the pen. Several techniques for selecting relevant points , i.e control points are discussed. We determine our best representation by weighting the criteria, analyzing the experimental results, examining the theoretical considerations and, making where necessary, justifiable tradeoffs. The techniques tested are objectively evaluated in terms of accuracy, efficiency and compactness.

The information characteristics (i.e signal entropy) of the handwriting and drawing signals, are estimated from the original and approximated data. Results are discussed and conclusions drawn. The research has been carried out on static data.

Preface

The work described in this thesis was carried out in the electronic division of the Open University Faculty of Technology, between October 1981 and March 1985 and the writing up has taken place since leaving the Open University Campus.

I should like to thank Prof John Monk for the use of the facilities of the Electronic Group, and I am also grateful to the technical staff of the Group for their assistance.

I am particularly indebted to Dr Gabriel Smol, who supervised this work, for his part in setting the project up, and for his invaluable and enthusiastic support throughout the project.

I am grateful to Drs Graham Read and Mark Cramping of the Faculty of Mathematics, who suggested a data reduction algorithm, and assisted me during its development and implementation. The algorithm is successful for hand drawn curves and has been published; (CRAMPING85).

I am pleased to acknowledge the financial support from the Open University. I shall always have the warmest memories of my time at the Open University Campus, Milton Keynes.

It is impossible to mention individually the many people who have helped me in this work; but the warmest memories of my late parents gave me the necessary support and encouragement to see this project through.

In accordance with the Open University Statutes, I hereby declare that no part of this thesis has been or being submitted for any degree or diploma at any other University, and that, except where indicated in the text, it is the result of my own work.

TABLE OF CONTENTS

CHAPTER 3: AVERAGE INFORMATION MEASUREMENTS OF HANDWRITING AND DRAWING SIGNALS FROM THEIR REPRESENTATIVE STATIC RAW DATA

Chapter 4: CONSIDERATIONS OF THE EFFECTS OF OF CONSTANT SUBSAMPLING RATE

Chapter 5: REAL TIME ALGORITHMS FOR THE REDUCTION OF POINTS REQUIRED TO REPRESENT FREEHAND PRODUCED DRAWING AND HANDWRITING

Chapter 6: SELECTION OF SIGNIFICANT PEN POSITIONS, BY METHODS WHICH MAY NOT WORK IN REAL TIME

Chapter 8: REVIEW AND COMPARISON

Chapter 9: Conclusions

LIST OF FIGURES

PAGE

LIST OF TABLES

PAGE

1.INTRODUCTION

1.1. Aims and Objectives

Since 1976, The Open University has been developing a system called " Cyclops " which is intended for distance education and training (READ77, FRANCIS81). It is based on the fact that microcomputers can process pictorial information in digital form; and the information can be stored on an audio cassette or transmitted along a telephone line, and transformed into a picture on a normal television set. Essentially, it implements a communication that enables the real time remote reproduction of hand generated information. The trajectory of the writing pen as produced at the input side is reproduced at the output side, including the effect of movement. Consequently, a hand generated message will appear on an output device as if it were written directly on that device.

By " message " we refer to free hand drawn curves; e.g. handwritten texts, drawing, diagrams.

From the telecommunication standpoint (Fig.1.1), the Cyclops system offers two combined input / output terminals, which allows both speech and diagrams to be transmitted; this is very useful during a telephone conversation because, one may make a point and enforce it by visual illustrations; it has been said " a picture is worth a thousand words ".

The pictures are displayed at both ends of the communication link. In a tutoring (or teleconferencing) situation, the tutor has a digitising pad which he or she can use as a blackboard. The tutor may also use a light pen. The student terminals also have light pens fitted for use on the screen. The same picture is displayed at both ends of a connection, both parties (i.e tutor, and students) can contribute to the same picture. A marker is available as an aid for pointing at a particular location in a picture.

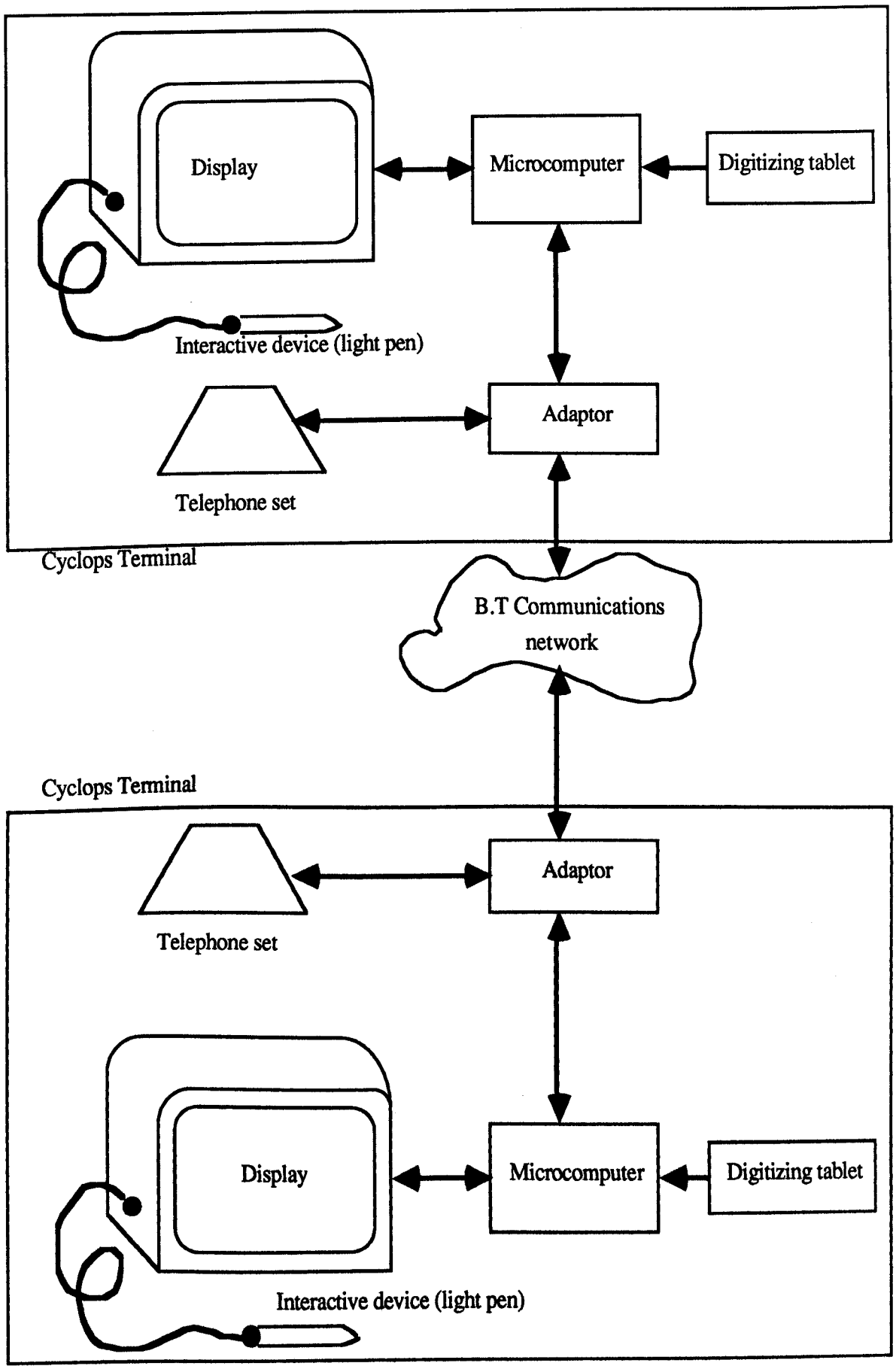


Fig.1.1 Schematic arrangement of a Cyclops System

The movements of a marker do not create a trace. It is possible to erase a whole picture, part of a picture or a specified trace. The traces may be of varying thickness.

The Cyclops system uses two telephone circuits per combined input / output; one for speech and one for graphics. This limits the usefulness of the system because of the expense of using two lines. It is therefore desirable that speech and graphic signals be transmitted simultaneously. To meet this requirement the graphic signal must be compressed. The compressed signal will then be combined with the speech signal and transmitted over a single telephone circuit. This will substantially enhance the usefulness of the " Cyclops system "; thus, in this thesis, our work has been carried out with a very practical engineering goal in mind.

Communication is to convey information generated at one point to another point. In most practical cases, these two points are connected via a channel which introduces noise. Modern communication theory starts with setting probabilistic models for the source which generates information and the channel which corrupts information. SHAN48 proposed a communication system in which information is processed both before and after being corrupted by the noise over the channel as shown in Fig.1.2.

The source encoder and the source decoder form a pair of elements which reduce redundancy present in the data to be conveyed and thus reduce the required speed of transmission. The channel encoder and the channel decoder, however, may also add redundancy to the source encoder output in such a manner that the redundancy added by the channel encoder helps the channel decoder to detect and correct some errors occurring over the noisy channel. Channel coding may also allow errors to be "avoided". For instance lines that reinforce the clock frequency component reduce the probability of errors due to loss of synchronization.

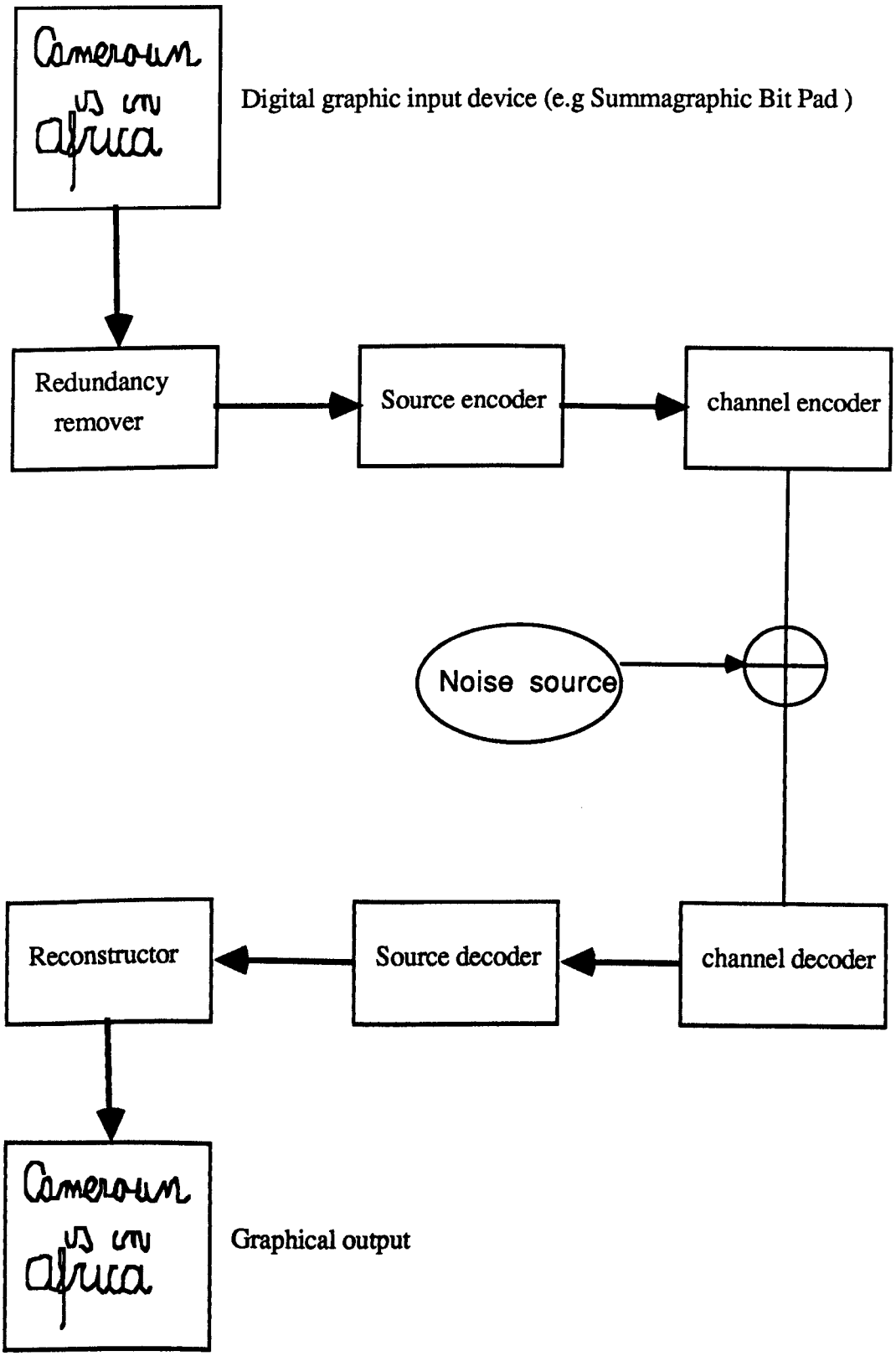


Fig.1.2. The block diagram for communication

With reference to Fig.1.2, the central theme throughout this thesis is the investigations into ways of approximating the graphic signal, so that after source coding of the approximated data signal, the bit rate is brought down to considerably less than 200 bits per second.

When the speech and graphics are combined for transmission, a practical graphic data transmission rate is 200 bits per second; this figure seems to be the baseline in work published in recent years (TOMIO83, LORIG81, MAKOTO80), presumably because one could extract the required bandwidth from the available speech bandwidth (300Hz to 3.4KHz), without significant deterioration of speech intelligibility. So it makes sense to have the approximated coded graphic signal bit rate to be considerably less than 200 bits per second.

To minimize transmission errors, redundancy must be built into the approximated coded graphic data source code. The addition of the redundancy will increase the approximated coded data signal to 200 bits per second. An obvious requirement in a redundant code is that it should allow error detection and correction at the receiver. The speech channel can accommodate 200 bits per second of graphic signal without significantly affecting the speech quality (TOMIO83).

The crucial point about hand drawn material is that the digital signal is produced as line segments, and that each sample of the signal is correlated to previous and subsequent samples. There is a high degree of correlation in the time sequence of signals corresponding to consecutive samples of the position of the pen on the digitising tablet, or the light pen on the T.V screen. The hand generated data "follows" line segments in time, as opposed to being derived from raster systems (e.g T.V and facsimile) which involve scanning of the source field, and which require comparatively large numbers of bits to be transmitted for each picture.

In the case of hand drawn graphic signals, fewer numbers of bits are needed because of the correlation between the consecutive samples in time; as opposed to two dimensional space in the case of scanning.
Most graphic compression work (HUANG67) has been done done on raster systems which have "lost" the temporal correlation inherent in any line segments. In our applications, the time dimension is of central importance.

Although our research arose out of a requirement of the Cyclops system, it has a much wider range of applications. There are many instances where the possibility of sending sketches over telephone lines would be useful;
e.g two engineers at different locations, discussing an engineering problem. One party may be interested in the storage of hand drawn pictures, or any picture which can be treated in terms of samples taken consecutively along its line segments. The line segments are essentially finer piecewise linear curves; approximating them by either coarser or smooth curves may lead to an economic storage.
Piecewise linear or polygonal curves have been used to approximate boundaries of graphical objects in cartography, typography, computer graphics, pattern recognition; they have resulted in substantial storage savings (KUROZ82, SKLAN80 and references there in).
In our work we evaluate some of these methods, in the context of hand generated graphic material, and identify those which are fast enough to be used in real time. In general hand generated material is made of curved sections and straight line sections; therefore it is useful to produce an approximating technique, which incorporates straight and curved parts; this requirement is necessary in applications where the approximated version of a drawn material is subject to special effects such as magnification. It is intuitively clear that , on the curved parts of the drawn material, a polygonal approximation technique might become unacceptable as the magnification

factor grows; for example the distortion of an artist's drawing would become unacceptable.

We have conducted experiments on the effects of magnification on hand drawn material approximated by straight lines; some illustrations will be shown in chapter 6. Under magnification, a curve approximated by straight lines shows kinks, i.e the transition from a fitted line to the next is unsmooth; this undesired effect has prompted us to consider cubic polynomial methods which ensure continuity and smoothness; this means that, assuming original smooth curve segments, the approximated curve sections of each curve segment should join one another without discontinuities and the tangents should be equal at the connecting points. Our experiments have shown that the smoothness is preserved under magnification.

This thesis does not attack the problem of communication and transmission. However, it is worth noting that, for communication, a protocol may be necessary. In the context of basic real time remote reproduction of " live " handwriting, and of teleconferencing the protocol requirements are fulfilled by the human beings participating in the communication.

If however the system caters for automatic transfer of messages between equipments, a more formal protocol is mandatory; (TANE81).

This thesis deals with the compression of the generated graphic data.

To carry out this, 13 one-hour electronics tutorials have been recorded, using a digitising tablet which is one of the peripherals of the North Star Z80 based data acquisition system developed by my supervisor; (SMOL81). The investigations were carried out on the database which represented a library of real tutorials. These investigations entailed :

One. Computer analysis of the informational properties of the signal associated with the original static graphic data.

Two. Computer analysis of the informational properties of the time or space subsampled data, obtained from various approximations.

Three. Comparisons in terms of graphic signal entropy rates.

Four. Discussions and comparisons of the approximation techniques in terms of storage requirements, approximation and reconstruction speeds. We also briefly consider the source coding of the approximated data produced by the best approximation technique. In this thesis, "best approximation" means optimum in terms of acceptable distortion, storage requirements, approximation and reconstruction speeds.

1.2 Background literature

Devices, enabling the transmission of short handwritten messages, using transmission of analog signals have been available for many years (PATENT73). However the transmission of hand generated data via digital signals started in the early ninety seventies.

The University of Delft (Netherlands) has been the major source of published information (BORD78). Recently, interest in handwriting processing has grown; research into the representation of hand generated material for transmission can be found in DEL80, TOMIO83, LORIG81, YUI82, and MAKOTO80. As far as coding is concerned, many authors seem to take their ideas from the work undertaken at Delft University. The general idea is that the sampling of a trace generally takes many bits. In practical situations, the amount of bits is reduced by differential coding techniques. The starting point of a trace is coded completely in x and y coordinates; the subsequent points of the trace are represented by coding the deviations with respect to the starting point. In this way, hand generated

information can be coded into a signal of at least 200 bits per second. We have found that the published methods are inefficient in terms of storage savings and signal bit rate. The hand generated shape representations used in many reported telewriting systems are very basic. Many employ arrays of points joined with straight line segments. Higher-order representations (e.g. at least second order polynomials) are not used.

We find very little information in the literature concerning the entropy measurements for handwriting and drawing signals. In this context, DEL80 has done some work; but did not look into how the signal entropy might be affected if non essential (i.e redundant) details were removed from the pen tracks data.

In this thesis, our work is devoted to finding ways to reduce pictorial information, by eliminating those aspects of the visual appearance of the picture that contribute little to the picture's interpretation or perception by a human observer. We show that appropriate efficient approximation methods, yield a signal bit rate, which is considerably less than the figure 200 bits per second, reported in most of the published papers.

1.3. Contents

Chapter 2 begins with a brief description of hand drawn data capture.
An overview of data compression is given, followed by the discussion of how the 200 bits per second graphical signal, can be combined with the speech signal.
Chapter 3 briefly recalls the relevant information theory, and deals with the statistical properties of the original data signal. In particular initial standard estimates of the nth order entropy, based directly on conditional probabilities, lead to results which are much higher than would be expected

intuitively. The explanation appears to be "unwanted detail" (e.g noise, wobble, stiction). This study calls for the reduction of storage and/or channel capacity requirements. This is achieved through the use of efficient data processing algorithms on the hand generated material. The desired algorithms should be automatic and should efficiently code hand generated material in real time, so that they can be reconstructed almost instantly at a distant receiving station.

Chapters 4 to 7 deal with smoothing digitised pen trajectory by data reduction techniques. On-line or real time data reductions algorithms are described in chapters 4 and 5. They receive one point at a time, and update the polygonal approximation accordingly, so that after points P_0, P_1, P_2, ... P_i, have been received, their polygonal approximation is available; this is the reason why, we have appropriately called them on-line. They can be used during the digitising as well as afterwards on the total data sets. Our own efforts in this field have lead to a recent publication (CRAMPIN85).

Off-line or non real time methods are dealt with in chapter 6 and chapter 7; they operate on the data collectively; in other words, information about all points of the set must be available before any of those algorithms can be applied.

The first of the approximation technique, described in chapter 4, is a decimation technique which simply deletes all but every nth point along the digitised hand drawn trace, where n is a fixed integer based upon the desired degree of approximation. The decimation technique (i.e subsampling) takes advantage of the sequential and time information which characterizes hand generated material from a digitising tablet. The alternative to deleting points is to select them; chapters 5 , 6 and 7 cater for this approach which tries to use only shape information. The informational bit rates of the approximated data are thoroughly discussed through chapters

4 to 6. Chapter 7 examines the problem of using cubic polynomials to generate smooth curves through an ordered set of planar points which represent handwriting and drawing. The trade offs between cubic and polygonal methods, applied to the reconstruction of segmented digitized handwriting and drawing curves, are analyzed. Throughout chapters 4 to 7 the error performance of various techniques is given both tabular and graphical form, a number of reconstructed drawings are shown for visual comparative purposes.

Chapter 8 presents the results of this research. Criteria for evaluating the coding techniques are reviewed. The results of the test runs made using the various techniques are then discussed, and the preferred methods and variations are selected. The best approximation methods to emerge from this research are thus identified. The efficiency has been evaluated in terms of storage, selection speed and how much error can be tolerated in the reconstructed picture. The results of entropy measurements are reviewed.

Chapter 9 contains the conclusions. It summarizes the best methods developed during this work, and discusses their relative merits. Analysis of the entropy measurements made is given. Finally, areas for further research are put forward.

This work contains one appendix. It contains the derivations and summaries of the formulas referred to throughout this thesis. Some criteria for evaluating mathematical descriptions of the pen trajectories are defined in the appendix; these are:

1. The accuracy of the representation.
2. The efficiency of the representation.
3. The compactness of the representation.

It is hoped that the repeated comparisons of entropy results with the target entropy (200 bits per second) will not irritate the reader. Those comparisons were seen as a guide on how well were doing in our quest for measuring limiting entropy estimate. Moreover no worker in the relevant field has published any entropy results less than 200 bits per second. Our ultimate entropy measurement was 35 bits per second. This represents a tremendous decrease of 82% with respect to the published figure.

1.4. Conclusions

The issues addressed in this introductory chapter, can be summarized in terms of :

1. What exactly is our work ? what are its attributes ?
2. Why is our work relevant ? what are its justifications ?
3. How is it possible to produce low bit rate handwriting and drawing signals ?

These three issues constitute the framework for the next eight chapters.

1.4.1 What is our work ?

In this thesis, we deal with the source coding (essentially removal of redundant information) of the hand generated material, so that the resulting graphical signal can be combined with the speech signal for simultaneous transmission over voice grade telephone lines, which are strictly band-limited to the range 300 Hz to 3.4 KHz. In our work, we do not deal with channel coding (essentially matching the signal to the transmission channel). Within the framework of hand generated material, the data "follow" line segments in time, as opposed to being derived from a raster sweep.

1.4.2. Why was the work necessary ?

As explained in section 1.1, the demands of the Cyclops system presented a practical motivation for our work. However we feel that the results of our work are of more general value because :

1. From our investigations, we find that the entropy estimates provide a target for coding. This is irrespective of the type of signal. So it does not apply only to handwriting and drawing signals. Thus for signal compression (i.e bit rate reduction) we can claim that finding the minimum entropy is a result of general scientific value.
2. Most graphic compression work has been done on raster systems, which have "lost" the temporal correlation inherent in any line segments. So we feel that our work is a contribution to knowledge in the field of compression of graphic signals characterized by temporal correlation.

1.4.3 The "How" of the work.

As noted before, the hand generated data "follows" line segments in time. One would expect a high degree of correlation between consecutive data points and this should lead directly to low limiting entropies, or failing that it should be possible to exploit the inherent correlation, in order to devise very effective coding, which is what we do in chapters 3 to 7.

To calculate the entropy estimates of the handwriting and drawing signals, 13 one-hour electronic tutorials were recorded. Those tutorials were either of the following types :

1. Handwriting only.
2. Drawing only.
3. Combination of handwriting and drawing, was the most common type of tutorial.

The written medium of communication was English language, but we expect similar results for any Latin based language. The graphic signal associated with hand generated material must be compressed to produce a target bit bit rate of at most 200 bits per second. The reason for this target bit rate is that the bandwidth it requires can be extracted from the available speech bandwidth (300 Hz to 3.4 KHz) without significant deterioration of speech intelligibility. The entropy estimates provide a target for coding. In chapter 3, we find that initial standard estimates of the nth order entropy (n = 8) based directly on conditional probabilities, lead to results which are much higher than would be expected intuitively. The explanation, appears to be "unwanted detail" (e.g noise, wobble, stiction). In order to test this hypothesis, we need to find some ways of fitting the curves so as to reduce them to parameters that contain all the information a Human viewer needs in order to perceive adequate detail, while leaving out all the unwanted detail. This representation contains all the relevant detail; we can therefore, use it to extract the nth order (effectively limiting) entropy for the appropriately coded signal. This lead to our final satisfactory result of about 35 bits per second.

A major spin off of our investigations is the development of suitable techniques for coding hand generated material and a measure of the effectiveness of these techniques is the entropy values they yield.

The rest of this thesis will attempt to answer the major question:

How can one measure the minimum entropy of handwriting and drawing signals ?

CHAPTER 2.1

2. HAND GENERATED DATA COLLECTION.

The conversion of free hand generated curves to computer readable numerical form is effected by a coordinate digitising tablet. The curvilinear representation scheme of the trajectory of the pen on a writing surface characterizes the data tablet. A digitising tablet converts the position of a hand held stylus into quantized coordinates of (x, y) values.

In order to obtain a picture as much detailed as possible,
our digitizer (i.e Summagraphics Bit Pad One) was operated at its highest sampling rate. This led to the large amount of data to be handled in different areas and to be transmitted from one place to another. Given this situation, we ask what data compression can do for us.

2.1 Introduction.

Graphic information results from moving a stylus on a writing surface. By graphic information, we mean a hand generated message. In the context of our work, a message refers to handwritten text as well as to drawings or diagrams; in fact any physical form which represents the trajectory of a pen moving on a writing area. In order to find out the statistical properties of hand generated messages, thirteen electronic tutorials (produced by different tutors) were recorded. A North Star Horizon microcomputer based data acquisition system, designed by SMOL81, was used to record the necessary data, which were transferred to a minicomputer computer (i.e VAX 11/750) for processing, analysis and interpretation.

The remainder of this chapter is structured as follows :

a. The representation of hand produced material at the input level.

b. A brief examination of the data capture system designed by SMOL81.

c. An overview of data compression in the context of digital processing of hand generated material.

d. A brief discussion of simultaneous transmission of handwriting and speech signals over a single telephone circuit.

2.2. Representation scheme at the input level.

We are dealing with digital processing of handwriting and hand drawing signals for communications. CATT69 specifies that the two basic concepts crucial in the digital processing of signals are sampling and quantization. Now, let us look at these concepts in the context of graphic signals associated with handwriting processes.

The trajectory of the pen is initially represented by an analog signal, which is uniformly sampled. Assuming that the resolution of the transducer system is adequate, the sampling process results in a time series of numbers which yield an accurate representation of the original signal; provided that the sampling rate is at least twice the bandwidth of the signal (i.e 2 f_m); f_m is the highest frequency contained in the waveform. Undersampling results in a phenomenom called aliasing, which introduces errors. However, CATT69 shows that the minimum sampling rate can be less than $2f_m$ if the lowest signal frequency contained in the waveform is not zero. Digital processing of the sampled data cannot occur until the data is quantized. Quantization of the time samples allows each sample to be expressed as a level chosen from a finite number of predetermined levels; each level can be represented by a digital number. After quantization, the analog waveform can still be recovered but not precisely; improved reconstruction fidelity of

the analog waveform can be achieved by increasing the number of quantization levels; unfortunately this requires increased transmission bandwidth; (CATT69).

A hand generated message is a curve or a succession of curves separated by pen lifts. One source of hand generated message is the operation of tracing along a specified path. The physical origin of handwriting or hand drawing is intrinsically analog and continuous in nature; thus we shall suppose here that the stylus position in two dimensions is given by the continuous functions x(t), y(t), and by the binary-valued function z(t), where t is a continuous, monotonically increasing ordering parameter. For our purpose t represents time. The function $z(t) = 1$ indicates a pen lift or a path traced solely to preserve the spatial relationship between otherwise non connected tracings; $z(t) = 0$ indicates that the electronic pen is down, subsequently the tracing data are valid. With reference to the above, the hand generated message could be the set of all tracings for which $z(t) = 0$.

Discrete sampling requires the traced path to be segmented. The segments in turn can be approximated by standard curves to which names can be assigned. Segmentation can be achieved by sampling the functions x(t), y(t), z(t) at regularly spaced increments in time; however this makes the size of the segments dependent on the tracing speed; (MYL72, ROG74). Alternatively, segmentation can be performed whenever the change in coordinates (x,y) from the last point of segmentation reaches a preset quantity; (NIEW73, FREE74, RAH80).

According to CATT69, the process of quantization involves three factors : the form of the quantisation, that is the rule according to which the quantisation takes place, the size of the quanta, and the approximant used to represent the quanta.

CHAPTER 2.4

Considering the theoretical background presented above, to sample and quantize our hand generated message, we are usually presented with two different approaches; raster and curvilinear :

Raster. In the raster approach, initially, one draws or writes on a scene area. The scene area is scanned over in television fashion, line by line, and the density of each spot on the scene area is recorded; (FREE74, DES78, SMOL76).

The spatial resolution of such spots is as fine as the transducer and the source medium permit. In effect, the analog medium scene is converted into a rectangular array of numbers. Each number is given the "grey-value" of the corresponding spot on the scene area. Usually an 8 bit quantity would normally suffice for each of the density values; (DES78). The raster method is not particularly efficient in terms of its use of digital storage media which include :

External memory; e.g tapes, disk.

Internal memory, e.g working space for data and programs, an example is R.A.M.

Usually, there is no need to actually transmit each spot value obtained from the raster scan. The strong correlation among neighbouring points in a scene may lead to extensive data compression that can be used to reduce the amount of data to be transmitted. The simplest scheme is the run length coding, where one indicates the particular density value at a point, and the length over which this density remains unchanged. Further compression can be obtained by using line to line correlation, or by using any of the more sophisticated image coding schemes described by NET80.

Curvilinear. In the curvilinear approach, the hand generated message is encoded directly, this is different from making a systematic raster scan over the entire scene area. A hand generated message consists of a curve of arbitrary shape. Each curve has a defined starting point and a defined finishing point. The starting point is when the pen goes down to begin a curve. The finishing point precedes the lifting of the pen from the writing surface.

A curve including starting point and finishing point may be called a trace. When one draws, or writes, the information is conveyed by traces on a contrasting background. The traces in a scene normally occupy only a small fraction of the total area; hence the direct encoding of the traces should yield a representation that is both more compact and more expressive of the information actually conveyed by the generated message suggests that a digitising input device which uses the curvilinear principle is efficient because it only generates the information that matters (i.e the descriptive information of a trace). A variety of graphic input systems using curvilinear representation have been developed (MYL72, ROG73, NIEW73,. SUM79).

For our work, we have used Summagraphics Bit Pad One; the details of its operation can be found in SUM79.

A digitising tablet (e.g Bit Pad One) describes the traces in terms of a series of digital coordinate pairs in the Cartesian system. The precision is limited by the smallest differences in X and Y that can be reliably detected in defining adjacent points of the curve. If a Cartesian coordinate system is used,

then all points used to describe the curve must be nodes of an implied uniform rectangular grid, oriented parallel to the coordinate axes and with spacing equal to the minimum detectable difference values. This is illustrated in Fig.2.1, where the minimum distances in X and Y for distinguishing between two nodes are labelled DX, DY, respectively.

Since virtually all two dimensional analog to digital conversion devices utilize a Cartesian coordinate system and possess uniform resolution, the uniform grid will almost always be the quantisation form that is used (ROG74, RAH80).

The minimum size is determined by the minimum resolvable coordinate difference. Since this is the same as the grid spacing there is no useful information available about the curve between two adjacent nodes. Hence such adjacent nodes can only be joined with the most primitive approximant, namely a straight line segment.

An entirely digital tablet (i.e Bit Pad One, SUM79) has been the most important input device of the data capture system which is described in the following section.

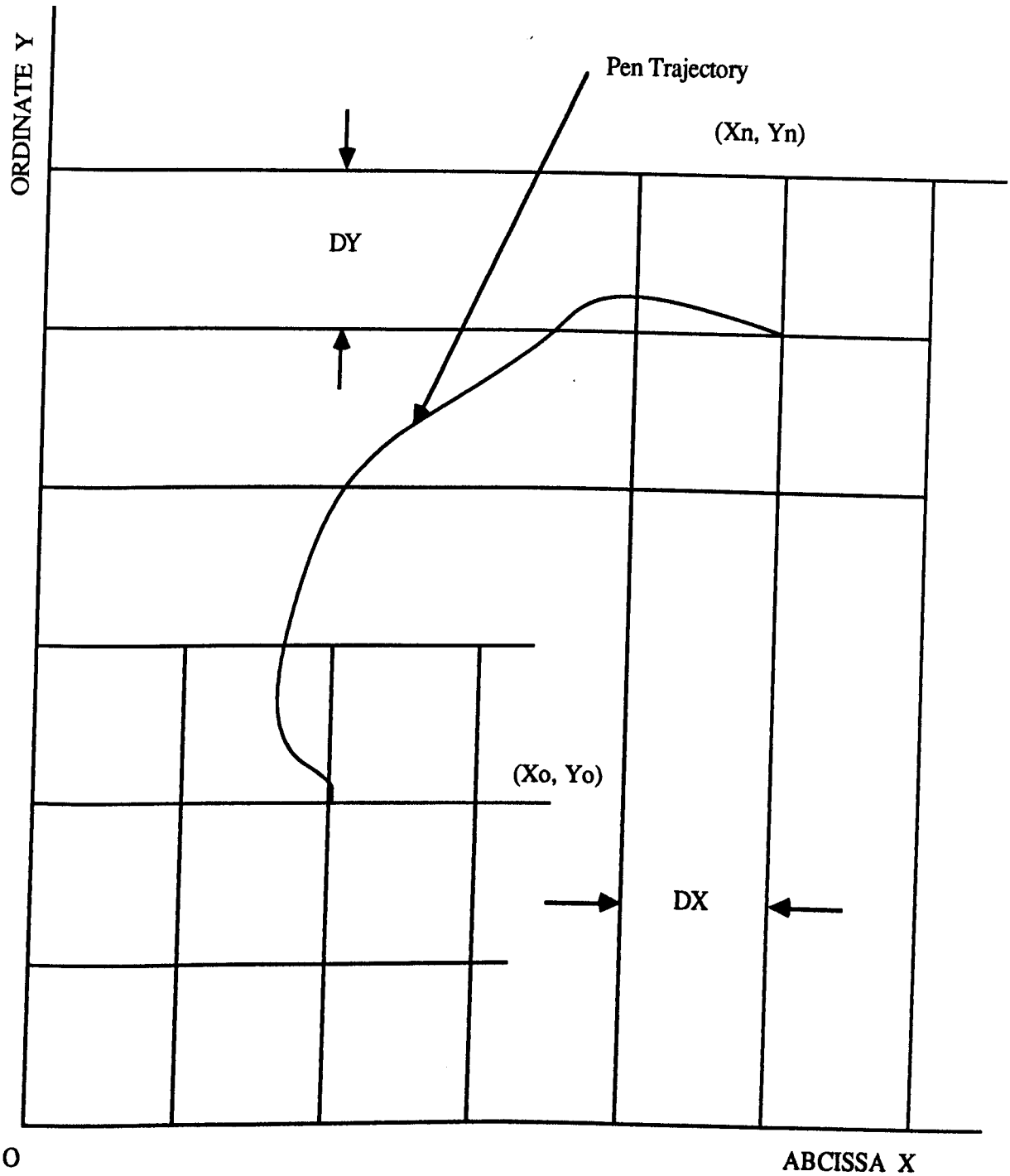


Figure 2.1 Implicit Uniform Square Quantization Grid

2.3. Data Capture

The data recording is best described with reference to Fig.2.2; it is organized as follows:

Hardware

North Star Microcomputer :

This is the nerve centre of the whole network, it is based on the Z80 microprocessor unit which controls the operation.

VDU Keyboard :

This element of the system allows alphanumeric inputs, and provides an input for commands necessary to change modes of operation. It enables the user to interact with the microcomputer with the help of the operating system DOS (Disk Operating System). Typical interactions are the loading and running of the application programs, such as PDMAKE, DM, VAXH (SMOL81), or transfer of data files to and from the floppy disk which resides in the disk drive, and listing of the files stored on the floppy disk.

MSBC-512 :

This is a card with variable resolution graphics. It is essentially a screen buffer, handled by the microprocessor; it is designed to interface a microcomputer to a TV type monitor. It has been programmed to display 512*512 dot raster. It drives the monitor, more details can be found in PERD81.

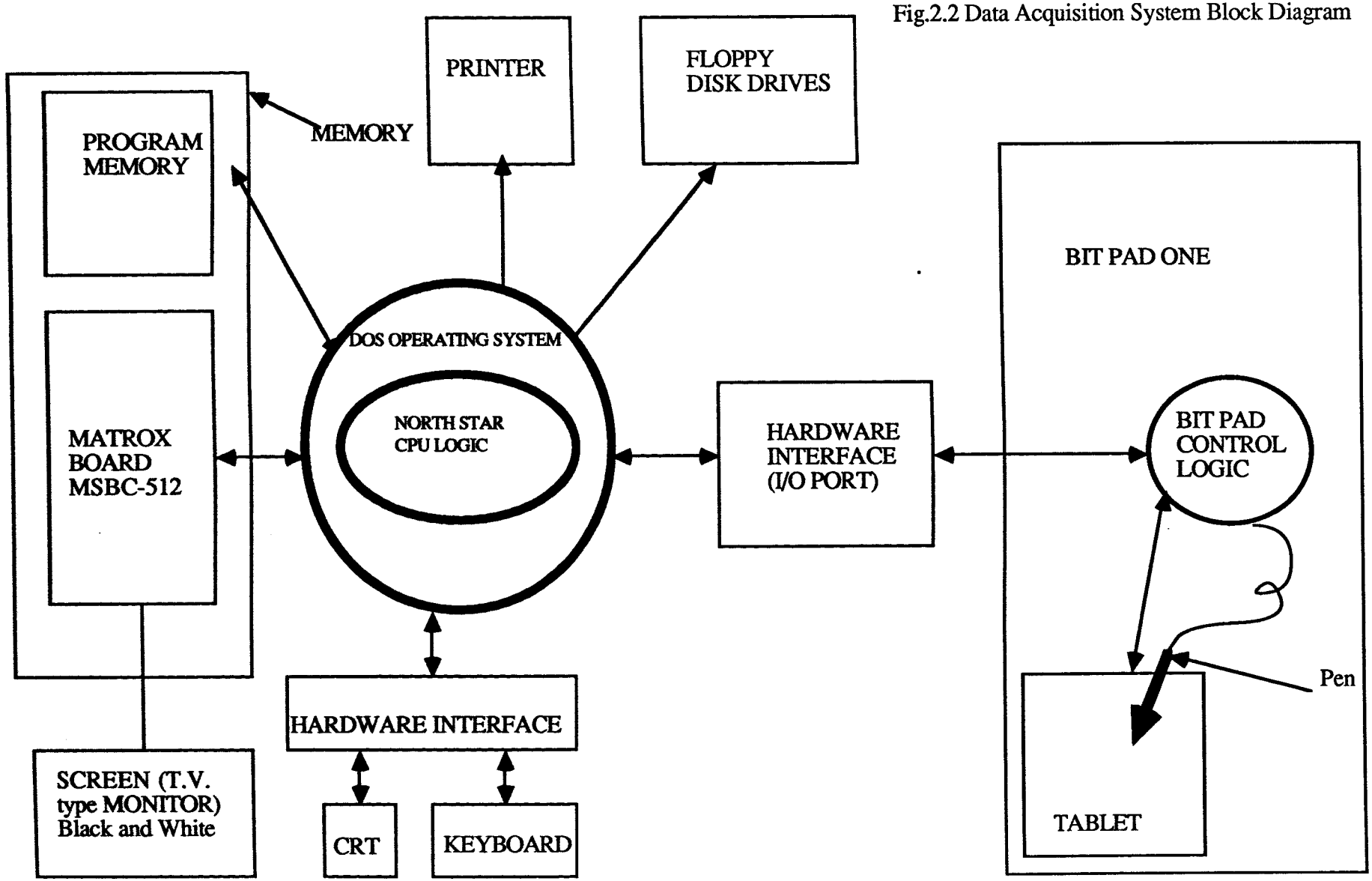


Fig.2.2 Data Acquisition System Block Diagram

Monitor :

This is a black and white TV type monitor (625 lines), only 512 lines are used and a line is divided into 512 dots. The screen buffer is mapped onto the monitor screen so that each point on the screen corresponds to one, and only one point on the buffer.

Summagraphics Bit Pad One :

This is a graphic data tablet, it is the most important input device. The Bit Pad one is essentially a square tablet that senses the position of an electronic stylus above its surface; this information is converted to digital information and sent to the microcomputer. The stylus with interchangeable marking tips is included with Bit Pad One.

Software

SMOL81 developed a set of application programs for handling handwriting data. We shall briefly describe only PDMAKE because it provides for the recording of data.

PDMAKE has been developed as an interactive application software for hand produced graphical data. It provides the interface between the Bit Pad One and the microcomputer. Essentially it produces a disk recording of the information drawn on the tablet.

PDMAKE uses a Summagraphic digitising tablet and the keyboard for the interactive input. Of the several options described in the Bit Pad One User's manual SUM79, SMOL81 chose to operate the Bit Pad One in stream mode rather than in point mode, or switch stream mode. The main point about stream mode is that it would allow an outline of a trace to be made quickly while the Bit Pad One transmits data continuously to the North Star Horizon Microcomputer. The 8 bit parallel interface has been used for the stream mode. In this mode of operation, the Pad transmits data to the microcomputer as a sequence of 5 bytes; this gives a total frame length of 40

bits. The acquisition of 40 bits of data takes place every 5 milliseconds; corresponding to the highest available sampling rate (200 samples per second) available with Bit Pad One (SUM79); thus from the point of view of data acquisition the bit rate is 8000 bits per second. When the user presses the pen hard enough against the Pad writing surface, a switch in the pen closes notifying the CPU of a "pendown action". As the user moves the pen across the tablet, the trajectory of the stylus is displayed on a monitor, at the same time the stream of coordinates and status data are processed and stored in the RAM space of the microcomputer internal memory, which is reserved for data; as soon as that space is full, the stored data are transferred to an appropriate external memory (eg floppy disk) for future use (e.g replay). The processing phases are as follows:

Phase 1. Compression of successive X-Y coordinate pairs.

Under static condition, the stylus is pressed down on the same position of the writing surface, this may create identical successive X-Y coordinates pairs. Under dynamic conditions, when the dimensions of the drawing are not large enough compared to the precision of the tablet, identical successive coordinates X or / and Y can occur.

Under dynamic conditions; i.e the pen tip moves on the writing surface, the distortion inherent to the discreteness of the tablet is at the origin of the sporadic, successive identical quantized magnitudes of the traces. These "successive identical values" should change but do not because the difference between the quantized and actual (analogue) coordinate values is less than the quantization level separation. This is usually known as quantization noise which appears when the dimensions of the drawing are not large enough compared to the precision of the tablet.

Should the pen be static, genuine successive identical samples would also occur.

A kind of "run length encoding" is applied to successive pen positions. It essentially maps pen position P_i to a sequence of pairs (P_i C_i), where P_i denotes pen position and C_i the number of times P_i has successively occurred.

Phase 2. Reduction of the number of quantizing levels.

Apart from the above compression, 9 bit accuracy coordinates representation were obtained by chopping off the three least significant bits of the 12 bits accuracy binary coded X, Y coordinates of a point. This is because the display resolution is 9 by 9 bits.

Phase 3. Data format for recording.

Under pen up conditions, no data bytes are stored. The 5 bytes transmitted by the Pad every 5 milliseconds are considered as garbage. However a pen up counter is increased every 5 milliseconds; and when the pen goes down to start a new trace, the pen ups counts are recorded. Before each new run (pen up or pen down) the equal coordinates counters are reset. Precise details of data format can be found in SUM79 and SMOL81.

It is worth while appreciating that PDMAKE sends to the monitor consecutive points which are distant from each other, in other words, identical successive points which are due to the quantization error, or to slow writing, or to the pen being still on the writing surface, are represented by one single point. So a basic filtering process takes place as to displaying points which build up the trajectory (e.g segment length average is about 80 data points); concisely a segment is defined as :

Pen down .....$X_{i-1}Y_{i-1}$ $X_i Y_i$ $X_{i+1}Y_{i+1}$.....Pen up

with (X_i , Y_i) = (X($i\Delta t$), Y($i\Delta t$)); where Δt is the sampling period.

The above is compressed as follows :

Pen down .....P_{i-1} P_i P_{i+1}.....Pen up;

with data set P_i = X_iY_i C_i; C_i is the equal coordinate count which accounts for how long it takes for the coordinate to change; taking into account the changes in coordinates (i.e ΔX, ΔY) the equal coordinate count contributes to an estimate of the stylus velocity in a dynamic situation.

The recording of data results in a file made of quantized handwriting samples; thus the original analog samples are not available; although they could be acceptably estimated by making use of the information (i.e equal coordinate counter) if the following conditions were met :

1. The signal bandwidth was good enough
   (e.g the highest frequency content of the signal).
2. The sampling rate was high enough
   (e.g at least twice the signal bandwidth).

The characteristics of data recording can be summarized as follows :

a) Because of the two-dimensional nature of writing, the sampling takes place for horizontal (i.e X) and vertical (i.e Y) direction simultaneously. Also a Z-information is present, namely pen up and pen down position. Within the concept of the trace, the pen down information is implied.

b) The use of an electronic tablet provides time sequenced online source data of a trace of the path of the tip of the writing instrument. The time sequence is accounted for by the equal coordinate count which tells us how long it takes for the coordinate to change; the unit of equal coordinate is one count per sampling period; and each period lasts 5

milliseconds. It tells us how fast the drawing is, for example when the drawing is slow successive digitized points are very close; they may even be identical in which case $C_i > 0$. From the spatial point of view, the equal coordinate count C_i tells us that the same coordinates X_i Y_i have repeatedly occurred C_i times; and this is essentially due, either to quantisation error created by the finite intervals of the digitising tablet, or to the stylus being held rock steady on the same spot.

c) In real time operation, there is effectively a match between the data points sequence and the operator's hand movement, interacting with the system, in the sense that the building up of a shape is synchronized with the dynamics of moving the pen stylus on the writing surface of the tablet. He may write (i.e the pen moves on the tablet), evaluate visually the response, think about it, and carry on writing. This typical sequence of handwriting processes leads one to define a hand generated message as a collection of strokes separated by pen lifts. A stroke is the series of points, from the first point where the stylus goes down to the last point where the stylus comes up. Stroke starting points have been identified in the point list because :

1. The user may move the stylus between strokes (e.g words)
2. The user may lift the pen within words; e.g to dot "i" or cross "t"; or even as part of his writing technique eg to write "it", the user may lift the pen to go from the end of the "i" to the top of the "t".

The identification has been accounted for by the (Pen down flag).
Once the point list is created, and stored, our data acquisition system has an electronic record of the hand generated information. This is really a form of communication which can be handled efficiently and consistently with other forms of electronic communication.

SMOL81's system was used to record graphic signals produced by tutors under genuine teaching conditions. Confronted with the problem of processing large amount of data for efficient storage and transmission, we ask ourselves the following question :

2.4. What can data compression techniques do for us ?

As pointed out, in chapter one, the two main reasons that call for an efficient compression of the " hand generated material data" are :

1. The need to store large quantities of data. This is important because, the same material may be used many times over again; this requirement is usually common in academic institutions (e.g Open University).
2. The desire to transmit digital hand generated material via telephones to students at a distant location. Here, we are not only concerned with with reduction of the data transmission rate for compatibility with telephone lines, but within the framework of Cyclops systems, our aim is to achieve a graphic bit rate, which makes it possible to transmit satisfactorily handwriting and speech signals simultaneously over a telephone channel. The problem of combining hand generated material and speech signals is discussed in a later paragraph.

The above implies that data compression can be defined as meaning any operation or transformation to reduce the amount of stored or / and transmitted data. An inverse transformation, data decompression, is usually applied to the recovery of the data which was originally stored or transmitted. Many methods and algorithms (IEEE80, IEEE81, IEEE85) have been studied, defined and applied to perform data compression operations. As a general classification they can be divided in two main groups:

1. Reversible methods, which permit, at least in principle, the recovery through decompression of all the original information.
2. Irreversible methods, which do not permit the recovery of all original data and which introduce some acceptable information loss or distortion.

Now, let us recall the theoretical aspects of data compression.

Fig.2.3 illustrates the mode of operation of a data compression system in a digital communication link.

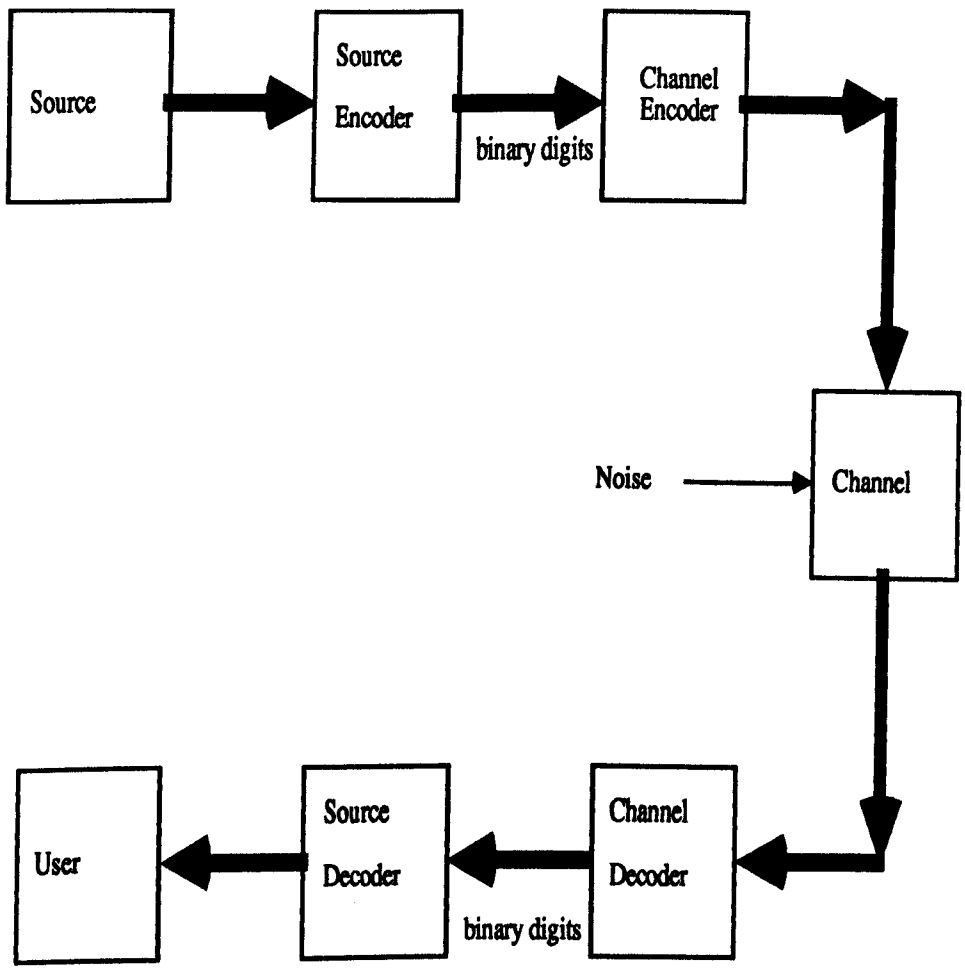


Fig.2.3 Block diagram of communication systems.

Earlier on, we noted that the Bit pad delivers data at 8000 bits/s.

Assuming start-stop data (2 stop bits, one start bit, one parity bit), for each 8 data bits, the effective transmission bit rate becomes 12000 bits/s. Clearly such a bit rate is prohibitively high to be accommodated economically over the switched telephones networks for the followings reasons :

1. Assuming that only handwriting data are sent down the line, the 12000 bits/s bit rate can be handled by todays high speed modems. But as the modem cost increases, with the maximum data rate it can cope with, it does not seem economical. As lower transmission rates usually translate into lower costs, it seems economically sensible to bring down the 12000 bits/s bit rate.
2. In our application, we desire to insert the graphic signal bandwidth into the speech bandwidth. In principle, the 12000 bits/s bit rate require 6000 Hz bandwidth, this more than absorbs the speech bandwidth which is 3100 Hz. The CCITT recommend a circuit responding to frequencies between 300 Hz and 3400 Hz as being adequate for the purposes of telephony, giving a high degree of speech intelligibility. Hence the justification of 3100 Hz speech bandwidth.

Considering the above, inserting the graphic signal into the speech signal is out of question because the telephone conversation quality will not be acceptable. So graphic signal bit rate reduction becomes mandatory if simultaneous transmission of handwriting and speech signal is envisaged.

To reduce the large number of pulses per second, and consequently the bandwidth, it is necessary, as already pointed out, to introduce data transformation represented by data or bandwidth compression. Such a transformation can be considered as one which operates on the data given by the Bit Pad, reducing the amount of non-useful or redundant data

and hence the bandwidth needed to transmit the required data through the telephone channel. The channel encoder in Fig.2.3 in the transmission chain represents the encoding of the compressed signal in such a way as to reintroduce suitable redundant bits to protect the encoded signal signal against the degradation introduced by the channel. The introduced redundancy is usually in form of error control coding, error detection and correction. Channel encoding is to be distinguished from the source coding, namely the transformation applied to source samples to obtain data compression. Within the context of this thesis, the encoding stage is viewed as in Fig.2.4

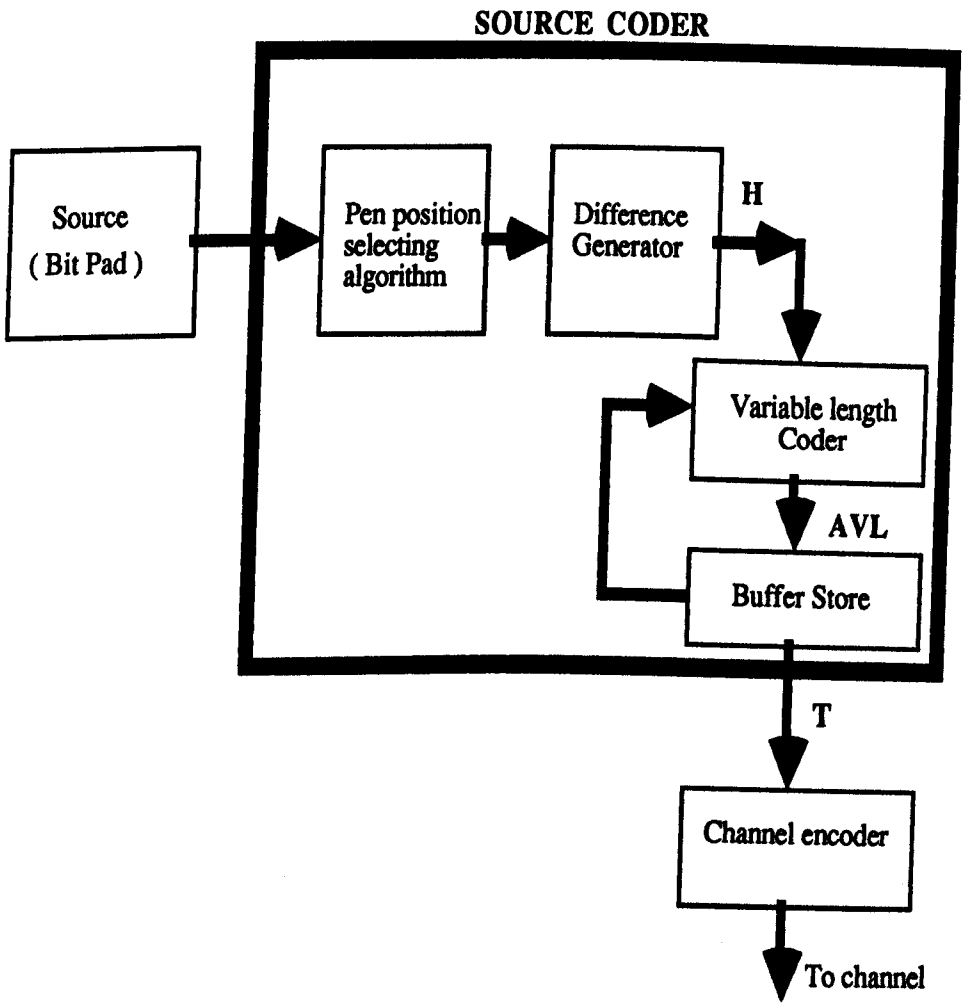


Fig.2.4 Components of the source encoding stage

Fig.2.4 indicates that in order to reduce the bit rate, the following actions must be taken :

1. Relevant pen positions must be selected from the successive pen positions generated by the Bit Pad (this is a data reducing process).
2. Difference transmission is fundamental to efficient source coding, because, for example the zero-order entropy of the difference signal is lower than the zero-order entropy of the full amplitude signal. So Bit rate reduction can usually be achieved by coding each difference independently (see next chapter for more details).
3. The difference signals should be subject to an optimal coding, e.g variable length coding. As the name implies, this type of coding generates a variable bit rate and so for a fixed bit rate system a buffer store is required to absorb this varying rate and to regenerate it a constant bit rate for transmission. We should note that the variable length coder must assume a representative set of difference probabilities.
4. In order to minimize buffer overflow or underflow, usually (CHOW71), there is a need for feedback from the buffer store to the variable length coder.

In Fig.2.4, H represents the entropy of the signal, AVL the average length of the code and T the transmission rate.

Channel encoding is indeed useful after source encoding because the compressed data are in general more sensitive to the communication channel noise than the non-compressed ones, due to the fact that at the receiver the signal is reconstructed using a lower number of samples. An error in the compressed data will generally introduce a considerable amount of distortion. For this reason it is necessary to protect the compressed data by using channel encoding (error control coding).

So far, we have put a lot of emphasis upon transmission; it is equally important to note that data compression can also be a very useful tool in local processing systems which do not involve any data transmission but where it is necessary to process or to store a great number of data or to set up information retrieval systems.

When we talk about data compression of samples from the Bit Pad, we mean coding them by a suitable code. Given the input (from Bit Pad) probabilities, it is of interest to determine the minimum number of bits required to code these inputs and generate a code that might achieve this minimum. In order to reach this objective we must understand the concept of entropy. As the entropy (SHAN48) concept provides a performance criterion against which we can measure a particular code, this brings us to give an " avant-gout" of the entropy concept, which will be explored further in chapter 3.

2.4.1. What is entropy ?

Given a Bit pad emitting the elements $m_1, m_2, ...m_q$ with the corresponding (independent) probabilities $p_1, p_2, p_3, ...,p_q$ the entropy of the source is defined by

$$H = -\sum_{i=1}^{q} p_i \log_2 p_i \qquad (2.1)$$

where log is taken to the base 2 for the measurement to be in bits.

In general, the entropy for q random variables can range from 0 to $\log_2 q$. The figure of $\log_2 q$ occurs when the source has a flat probability distribution, i.e $p_1 = p_2 = p_3 = p_4.....= p_q = 1/q$. The figure of 0 occurs if an element of a source is certain to happen (e.g $p_q = 1$) and all the

other elements have a zero probability, i.e $p_i = 0$ for i $1 \leq i < q$.

Entropy is a measure of the degree of randomness of the set of random variables. The least random case is when one of the random variables has probability 1 so that the outcome is known in advance and H = 0. The most random case is when all events are equally likely. In this case $p_i = 1/q$ for all elements a source (i.e $1 \leq i < q$) and $H = \log_2 q$. This concept is similar to the entropy concept in thermodynamics.

In our applications, entropy represents the amount of information associated with the set of coder input values and gives a lower bound on the average number of bits required to code those inputs.
The entropy defined in (2.1) is the first order entropy. It takes into account only the relative probabilities of the q possible Bit pad emitted elements $m_1, m_2, ...m_q$. If successive elements are independent, then the first-order entropy is also a bound on the average number of bits per element required to code a sequence of elements. If successive elements are not independent then the entropy (called H_2) associated with a sequence of elements is less per element than for an individual element. H_2 is the entropy taking correlations of order up to 1 into account. Similarly, one can talk about a third entropy which is the entropy taking correlations of order up to 2 into account. In general H_n is the entropy taking correlations of order up to n - 1 into account. This is explored in depth, in the next chapter.

Several properties of the entropy function H in (2.1) are described in ABR63 and JEL68. To underline one of these we may introduce the concept of the wordlength of the source code word.

If we design a code with code words $c_1, c_2,.....,c_q$ with word lengths $l_1, l_2,.....,l_q$ the average number of bits required by the coder is

$$R = \sum_{i=1}^{q} p_i l_i \qquad (2.2)$$

where p_i is the probability of the element i of the source.

If R is close to H the coder is near optimum; if it is significantly different from H it is not. A code will be said to be compact for our particular source if its average length is less than or equal to that of any uniquely decodable code. Searching for compact codes means that R must be made as small as possible. From (2.1) and (2.2), it can be proved that (JEL68) that

$$H \leq R \qquad (2.3)$$

According to (2.3) the entropy of the source is the lower bound for the code average length, and the ratio

$\mu = H / R$

is defined as the efficiency of the source code, while $(1 - \mu)$ is the redundancy.

A data compression method can be therefore considered as a transformation of the source data with an average length R as near as possible to H.

There are methods of developing compact codes (IEEE85). The most celebrated one is due to Huffman (HUFF52). In Huffman codes, the length l_i of each code word c_i is inversely related to the probability p_i. In this way the more probable and therefore the more frequent words will be encoded in shorter sequences compared to the less probable ones.

Equation (2.1) can be used to evaluate an upper bound for the mean compression ratio :

$$C_r = L / R \qquad (2.4)$$

where L is the average wordlength of the untransformed source, and R the average wordlength of the transformed source. L is a constant value due to the specific source (40 bits in our case), therefore the maximum value for C_r is obtained for R minimum. So from equations (2.3) and (2.4) we have

$$\text{maximum value of } C_r = L / H \qquad (2.5)$$

It is important to remember that the equations (2.3), (2.4) and (2.5) depend on the hypothesis of perfect reversibility of the compressed signal, which is expanded to give back the original signal with no errors.

A higher value for C_r than that expressed by (2.5) can only be obtained by introducing a certain amount of distortion in the reconstructed signal.

This is, what we call, approximate coding methods which are found in later chapters (4, 5, 6, 7). Approximate coding methods introduce distortion to the reconstructed picture. In the context of our work, they operate as follows:

As the original curve is drawn, relevant points are periodically selected and sent along the communication medium. At the receiving end, these points are used to reconstruct the curve through an interpolation process,

e.g first or higher order polynomial interpolator .

To recapitulate, should perfect retrieval of the original data be necessary, reversible data compression algorithms would be required. However, in our application, perfect reproduction of the source signal is not required because we want reproduction that is "good enough" for the user - this may not be exact reproduction.
Thus we can accept a certain number of errors; in order to increase the compression ratio. Approximate coding techniques (i.e irreversible), which enable us to reconstruct the signal with sufficient fidelity, give a higher efficiency.

In the following chapters, we make use of the fact that the redundant information in a digital signal arises from one or both of two reasons (RIST73) :

1. Neighbouring signal samples are not statistically independent.
2. The quantized signal values do not occur with equal probability.

The redundancy resulting from these two different reasons can be partly removed by separate techniques which may be employed concurrently to achieve a large data compression ratio.

Assuming that an efficient data compression of hand generated material, leads to a suitable average graphic signal bit rate (e.g 200 bits per second or less), it is possible to combine simultaneously speech and handwriting signals. A 200 bits per second graphic information is the standard bit rate (CCITT85) because a very small part of the telephone frequency band, the voice band, can be reserved for the transmission of the pictures without any unfavourable influence on the telephone conversation. CCITT85 estimates that the transmission speed of the graphic material corresponds with normal writing speed.

A very low graphic signal bit rate, i.e very much less than 200 bits per second will mean :

1. Improved speech intelligibility.
2. Increased complexity of the design of the analog bandpass filter, which is needed to extract a band segment (reserved for the graphic signal) from the speech band.

The combined signal (handwriting and speech signals) can be transmitted over a single telephone circuit. But how is this combination done ?

2.5 Brief discussions of simultaneous transmission of handwriting and speech signals over a single telephone circuit.

Speech signals and handwriting (and drawing) signals are required not to interfere during transmission. At the transmitting station, the following methods may lead to the prevention of interference.

Method 1.

One can block the whole microphone signal during graphic signal (i.e handwriting and drawing) transmission. It is possible to have a voice detector to inform the main processor, when the operator speaks. The processor then finishes transmitting its current graphic data and operates the switch which links the microphone to the line.

Implementation details of this approach can be found in ISHII79.

A standardized modem (e.g V21, ITU85) can be used for graphic transmission. V21 modem uses FSK modulation, which is a general technique of switching on or off oscillators in accordance with the graphic signal bit stream (NCC82).

V21 modem can be used at low data rates (e.g 200 bits/s).

Method 2.

Consider the spectrum of CCITT V21 modem (Fig.2.4).

Here, we choose channel 2, for transmission of graphic data, where a logical 0 is converted into 1850 Hz, and logical 1 is represented by 1650 Hz. The centre frequency is 1750 Hz, thus we can talk about a frequency deviation of 100 Hz in the positive direction (for 1850 Hz) and the negative direction for (1650 Hz). Considering that the highest frequency content of our 200 bits per second graphic signal, is 100 Hz, using Carlson's rule (SMOL76) the band segment required is 400 hz ranging from $1750 - (D + f_m)$ to $1750 + (D + f_m)$ where the frequency deviation $D = 100$ Hz and the highest frequency content of the signal is $f_m = 100$ Hz.

So the spectrum taken up by the graphic extends from about 1550 Hz to 1950 Hz. So we can block this frequency band which is needed for graphic transmission. In other words, a window (1550 to 1950 Hz) is reserved in the speech bandwidth (300 - 3400 Hz) to insert the data corresponding to the handdrawn or written information.

To recapitulate, the coupling of the writing signal to the speech is done by means of a two way "bandpass - band stop" filter. The stop band in the speech channel is from 1550 to 1950 Hz. The pass band of the writing signal is from 1650 to 1850 Hz.

Method 3.

In AKIYAMA83, we note that applying voice interleaving technique can successfully produce simultaneous voice and handwriting signals transmission. BRADY68 reported that, in a telephone conversation, a transmission channel does not contain any voice signal for about 25 percent of the holding time in average. Thus other signals may be

transmitted during the silent period.

If the bit rate of the handwriting signals is reduced to 200 bits/s, this rate is so small that simultaneous transmission can be attained by employing the voice interleaving technique because the loss of speech intelligibility is acceptable.

Details of this method can be found in AKIYAMA83.

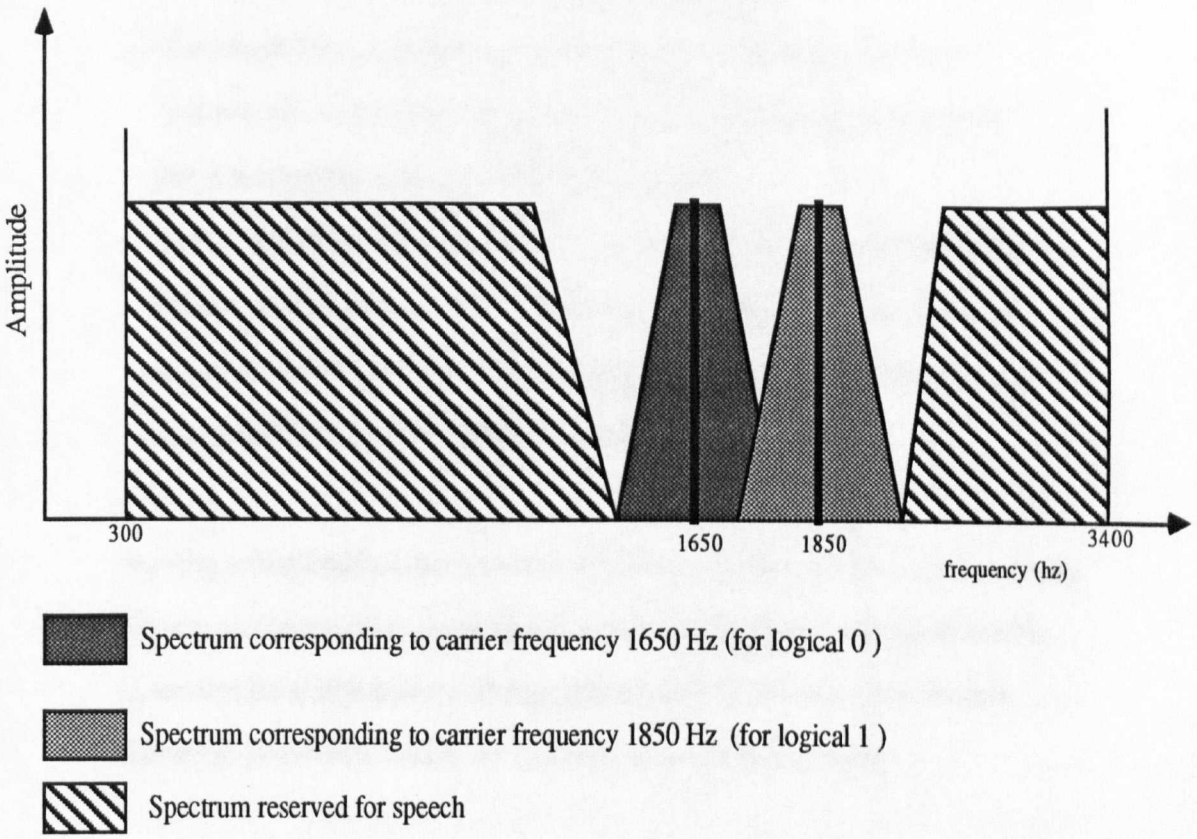


Fig.2.4 V21 modem modulation in practice, only channel 2 frequency spectrum is shown.

2.6 . Conclusions.

The input representation and data capture of digital handwriting data have been described. From this chapter, we know that handwriting consists of curves of arbitrary shape. Each curve has a defined starting point and a defined endpoint. Such a curve, including starting point and endpoint is called a trace. In our work, we are motivated by :

1. The need to reduce storage requirements for traces.
2. The interest in techniques that achieve maximum reduction in the quantity of " trace data " to be transmitted, subject to the constraint that a reasonable amount of data be preserved.

In the following chapters (4, 5, 6, 7), we place emphasis on reducing the amount of data that must be transmitted. In this respect the encoding technique needs not be information-preserving, as long as the resultant hand generated material is acceptable for visual analysis.

Having transferred all the raw data files to the minicomputer, we can easily operate on them using appropriately written utility programs which enable us to explore many aspects of digital handwriting signals, such as their statistical properties, which are considered in the next chapter.

3. AVERAGE INFORMATION MEASUREMENTS OF HANDWRITING AND DRAWING SIGNALS FROM THEIR REPRESENTATIVE STATIC RAW DATA

From the previous chapter, we know that successive traces (i.e pen tracks) separated by the pen lifts, define handwriting and drawing. In this chapter, the entropy or average information of the handwriting and drawing signal is estimated from a database, which is made of the static raw data tracks. The results of the entropy estimates call for smoothing of data.

3.1 Motivations

From chapter 2, it is known that the data capture rate is 200 pen positions per second. Five bytes are allocated to each position. If the signalling rate is identical to the sampling rate, then for asynchronous transmission requiring two stop bits, one start bit, and one parity bit for each byte, the bit rate ensuing for a real time transmission of the digital handwriting and drawing signals is 12000 bits per second. In principle, 6000 Hz bandwidth is needed for 12000 bits per second, so this is prohibitively high for the available standard telephone lines, and thus bit rate reduction becomes mandatory. Methods by which the bit rate may be reduced are essentially bandwidth reduction techniques which depend upon information theory; that is, these techniques attempt to exploit statistics of the signal so as to remove redundancy and to code the signal in such a way that the bit rate is reduced while at the same time the trajectory of the pen presented to the observer is essentially unchanged. Methods depending on the statistics of the signal must have as their basis a detailed knowledge of the statistics of the signal.

A potential figure of merit for evaluating a statistical coding technique is the upper bound (HAM80) of the entropy rate of the relevant features contained

in the data source. Information has the property of reducing uncertainty of a situation. SHAN48 defines the entropy as the measurement of uncertainty. If entropy is large, then a large amount of information is required to clarify a situation; if entropy is small, then only a small amount of information is required for clarification.

In this thesis, the entropy rate is expressed in terms of bits per second. Although we may talk about the " entropy per point " which contributes to finding the ultimate entropy of the signal, the notion of a point only arises because of the sampling, which is not a property of the signal. We have to deal with points because that is the way the data is presented to us, but these points are not relevant to our ultimate definition of entropy. At all times the reader must be aware of the distinction between bits per second, i.e the fundamental quantity, and bits per point, i.e the way the sampled data comes to us. Writing is essentially analogue and generated as n bits per second. Digital sampling leads to bits per point and bits per second, which equal the number of bits multiplied by the number of points per second. The number of bits per second is then the first estimate of the source entropy rate.

The remainder of this chapter is structured as follows :

1. Section 3.2 deals with Problem Statement.
2. Section 3.3 discusses the estimates of the entropy rates.

3.2 Problem Statement.

In 1960 A. Reny said (BERLEK74):

" The fact that information can be measured is by now generally accepted. How and with what expression of measure this should be done, is not an open and shut question. We intend to deal with this question ".

How do we reflect Reny's thoughts in the context of quantized handwriting and drawing signals ? To start with, we have to decide about the data set which should be used to estimate the entropy rate of the quantized handwriting and drawing signals.

3.2.1 Which data set should be extracted from the original and why ?

From chapter 2, it is known that a set of hand generated data was collected by operating the Bit Pad One at its maximum sampling rate, 200 samples per second with 9-bit resolution per sample (either coordinate x or y).

The signal associated with hand drawn material is produced as line segments, and each sample of the signal is correlated to previous and subsequent samples. Intuitively speaking, there is likely to be a high degree of correlation in the time sequence of signals corresponding to consecutive samples of pen coordinates which define the trajectory of the stylus on the writing surface of a digitizing tablet. One can markedly reduce the inter-sample correlation by differencing the signal. This correlation reduction was verified by evaluating autocorrelation functions on coordinate (x, y) samples and on the Nth (N = 1, 2) order differences of the sample data.

The autocorrelation coefficients for x(t), y(t) were estimated respectively from the usual expressions (OPENU75).

$$\emptyset_x(k\Delta t) = \left(\sum_{i=0}^{i=n} x(i\Delta t)x((i+ k)\Delta t) \right) / (n + 1) \qquad (3.1)$$

$$\emptyset_y(k\Delta t) = \left(\sum_{i=0}^{i=n} y(i\Delta t)y (i+ k)\Delta t) \right) / (n + 1) \qquad (3.2)$$

where k varies from 0 to n, n + 1 is the total number of samples representing the signal under consideration, i is the sample number; Δt is the sampling interval; k is associated with kth delay; and the above expressions generate the kth autocorrelation coefficient of signals. Appropriate coefficients are found by normalizing equations (3.1) and (3.2); this is done by dividing all results produced from (3.1) by $\emptyset_x(0)$ and all results produced from (3.2) by $\emptyset_y(0)$. The ensembles $\{ \emptyset_x(k\Delta t), \emptyset_y(k\Delta t);$ $k = 0, 1, 2.....n \}$ obtained from equations (3.1) and (3.2) define the autocorrelation functions for signals x(t), y(t); the autocorrelations functions for difference signals are carried out in the same way.

Discussion of the results:

The measurements of the autocorrelation coefficients of digitized signals x(t), y(t), $\Delta x(t)$, $\Delta y(t)$ are shown in Table 3.1, where delay is expressed in units of sampling period (i.e 5 milliseconds). Fig.3.1.a shows the autocorrelation function of horizontal signal x(t). Fig.3.1.b shows the autocorrelation function of first difference of signal x(t). Fig.3.2.a, Fig.3.2.b show the autocorrelation function of signal y(t), and first difference of signal y(t). A visual inspection shows that the autocorrelation functions of x(t), and y(t) are almost identical, they follow a second order polynomial trend, i.e if quadratic polynomials are fitted to the

autocorrelation coefficients, the maximum deviation between the fitted curves and the experimental curves may be quite small.

A study of columns 2 and 3 (Table 3.1) clearly suggests that each sample of the signal is correlated to previous and subsequent samples. The correlation is still quite pronounced even when the delay is 25 milliseconds. The autocorrelation functions of $\Delta x(t)$, and $\Delta y(t)$ follow rather an exponential trend, the coefficients decrease sharply with increasing delay. The autocorrelation becomes less and less pronounced, over 3 samples ; i.e 15 milliseconds. Fig.3.1.b and Fig.3.2.b clearly illustrate a reduction in the variance and correlation between samples; thus we have found that the difference signal sequence is much less correlated than the original signal sequence. The average difference in amplitude between successive samples is smaller than the average magnitudes samples themselves. Given the above discussions, the set of first order forward differences derived from the recordings of 13 one hour tutorials (approximately 1283975 samples) became the actual set of source words for assessing entropy rate estimate of handwriting and drawing signals.

Results of autocorrelation coefficients meaurements				
Delay	signal x(t)	signal y(t)	signal Δx(t)	signal Δy(t)
0	1.000	1.000	1.000	1.000
1	0.997	0.997	0.743	0.762
2	0.988	0.988	0.556	0.519
3	0.973	0.973	0.413	0.398
4	0.952	0.952	0.288	0.286
5	0.925	0.924	0.210	0.199
6	0.891	0.890	0.165	0.158
7	0.851	0.851	0.120	0.113
8	0.804	0.800	0.099	0.105
9	0.751	0.745	0.075	0.061
10	0.692	0.683	0.086	0.055
11	0.627	0.616	0.056	0.049
12	0.558	0.545	0.048	0.057
13	0.484	0.470	0.037	0.023
14	0.407	0.393	0.039	0.011
15	0.326	0.313	0.028	0.017
16	0.245	0.232	0.023	0.007

Table 3.1 Autocorrelation coefficients of x, y, Δx and Δy

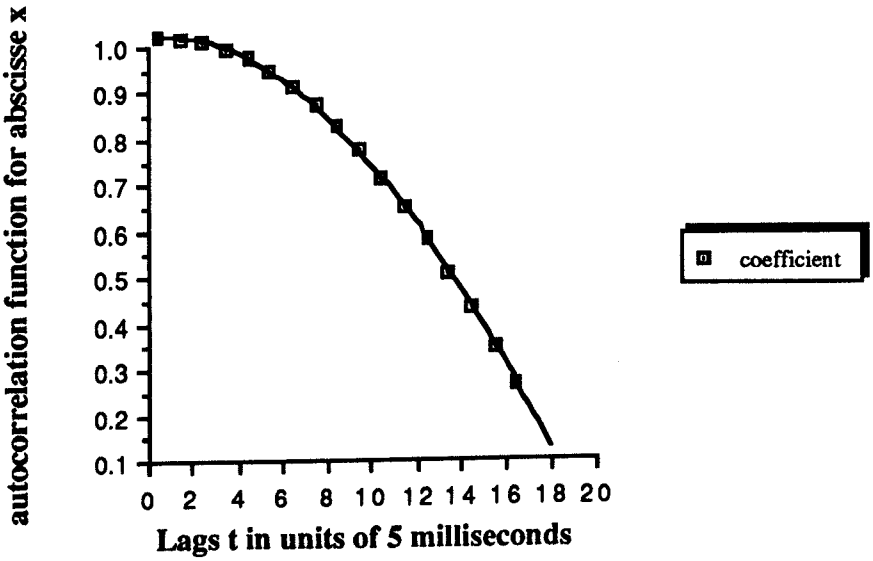


Fig.3.1.a

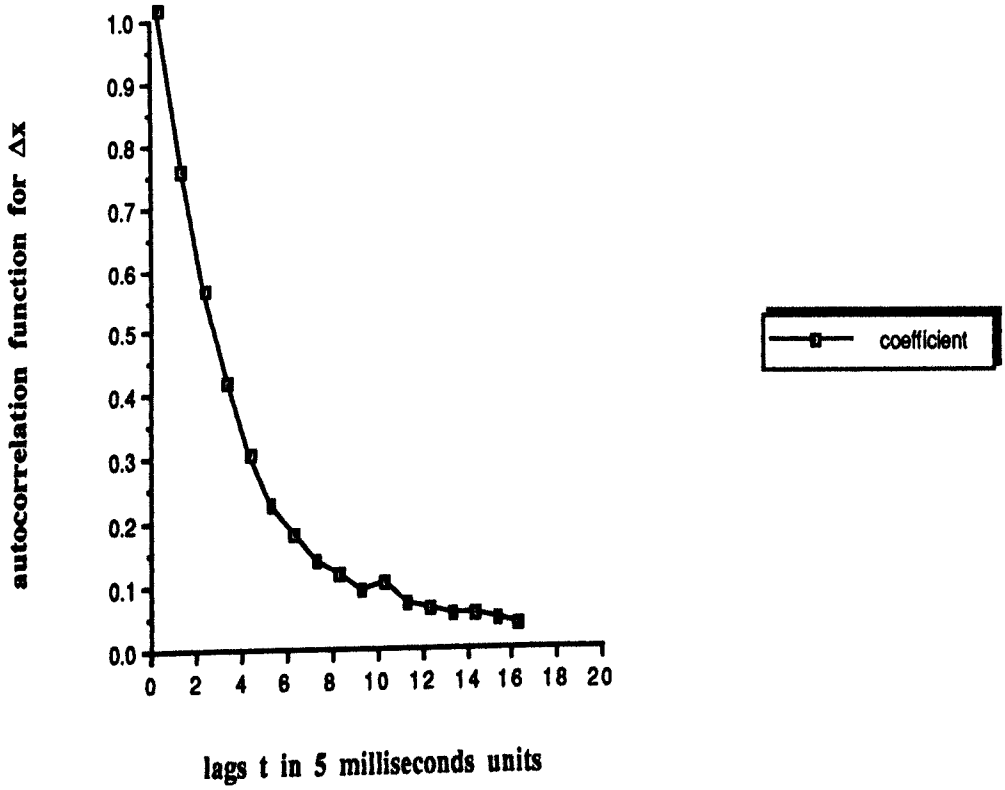


Fig.3.1.b

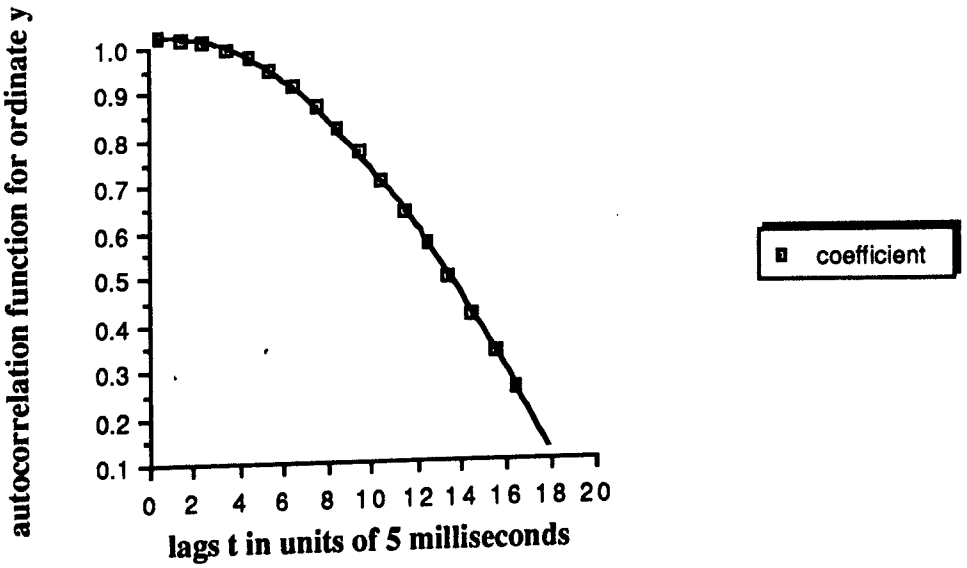


Fig.3.2.a

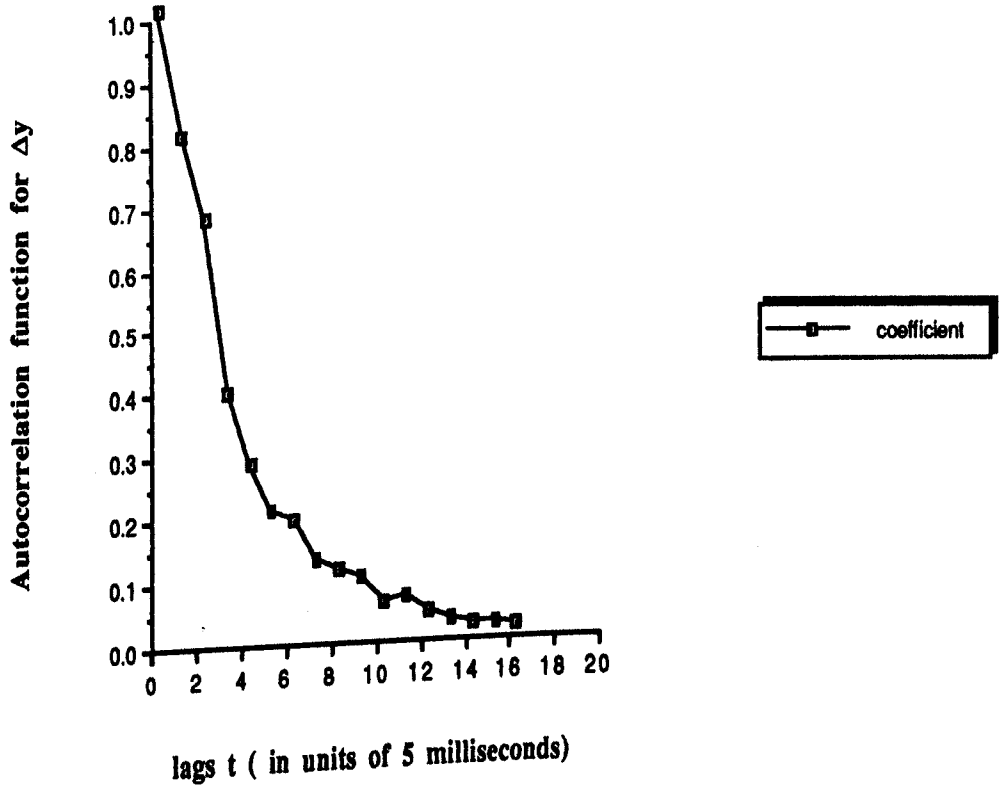


Fig.3.2.b

3.2.2. Significance of the entropy rate

The theory of the entropy of a source is a well established concept (HAM80, SHAN48). It is assumed that our source is characterized by first order forward difference signals represented by coordinate differences for successive pen positions. The relative difference technique transforms the full coordinate values into difference values, and generates a new source alphabet whose elements are the pen displacements (i.e movements).

If the pen moves from position (X_{i-1}, Y_{i-1}) to position (X_i, Y_i) a pen displacement is defined as :

a. $(X_i - X_{i-1})$, if only abscissa x is considered.

b. $(Y_i - Y_{i-1})$, if only ordinate y is considered.

c. $(X_i - X_{i-1}, Y_i - Y_{i-1})$ if x and y are considered simultaneously.

Let our source alphabet be made of K classes of displacements $\Delta_1, \Delta_2, \Delta_3, \Delta_4,..\Delta_i,...\Delta_K$ associated with probabilities $p(\Delta_1), p(\Delta_2),...p(\Delta_i),...p(\Delta_K)$. If the successive pen displacements take values independently at random, and if the probability of occurrence of the i'th class is $p(\Delta_i)$, then the information conveyed by a pen movement assuming the ith class is $-\log_2 p(\Delta_i)$. The average information conveyed by each class in the sequence is the entropy of the distribution given in the following equation

$$H_1 = - \sum_{i=1}^{i=K} p(\Delta_i) \log_2 p(\Delta_i) \qquad (3.3)$$

The average information has its maximum possible value when each class is equally probable with probability 1 / K.

Now if successive pen movements occur, not independently, but with

statistical influence extending from past movements, the average information per pen movement is less than that given in equation (3.3). Considering a state representing a particular past history, i.e, a set of known past Δs in the sequence, the average information per pen movement is given by :

$$H_n = -\sum_{j1\,,\,j2\,,\,j3\cdots\cdots jn\,,\,j} p(\Delta_{j1}, \Delta_{j2} \ldots\ldots \Delta_{jn}\,,\, \Delta_j)\ \log_2 p(\Delta_j \mid \Delta_{j1}, \Delta_{j2}\ldots\ldots\Delta_{jn}) \qquad (3.4)$$

where $\Delta_{j1}, \Delta_{j2} \ldots\ldots \Delta_{jn}$ denote a sequence of n pen movements Δ's. Although this equation was taken from HAM80, it can be found in most textbooks on information theory.

Assuming a stream of symbols, the expression

$p(\Delta_j \mid \Delta_{j1}, \Delta_{j2}\ldots\ldots\Delta_{jn})$ is the conditional probability that, given the occurrence of the state $(\Delta_{j1}, \Delta_{j2}\ldots\ldots\Delta_{jn})$, the next pen movement will assume the jth Δ; $p(\Delta_{j1}, \Delta_{j2}\ldots\ldots\Delta_{jn}, \Delta_j)$ is the joint probability that state $(\Delta_{j1}, \Delta_{j2}\ldots\ldots\Delta_{jn})$ occurs and is followed by a pen movement assuming the jth Δ. Equation (3.4) yields an estimate of n'th order entropy per pen movement. An n'th order probability distribution permits the calculation of an nth order upper bound on the information content of the signal. It provides a lower bound to the maximum compression. This means that, should we use an efficient coding system which utilizes just the statistical relation among the successive pen positions, the best we could possibly do is at least as good as an upper bound to the lowest data rate achievable by using compression. If, in fact, a significant statistical relation existed among say five such positions, we would want to measure a fourth order distribution. If we measured a lower order distribution we would then calculate a looser (higher) upper bound on the information content, and

would be more pessimistic than necessary about the prospects for the bit rate reduction, and hence of bandwidth reduction. The statistics worked on ought to be reasonably uniform for the range of the subjects' handwriting likely to use the system. Assuming that our data base is truly representative of all usage, the n'th order distribution will actually give us a great deal of information. It will for example, suggest whether the generation of handwriting and drawing signals (associated with data obtained from electronics tutorials in English) is a (n-1)'th or nth order statistical process. Putting it in terms of correlation distance, we are effectively saying that $p(\Delta_j \mid \Delta_{j1}, \Delta_{j2}....\Delta_{jn}) = p(\Delta_j)$ for n above the correlation distance, e.g samples more than n away do not correlate. For example, if the only statistical influence which exists in a handwriting / drawing process is between 5 successive points, then the fourth order approximation to the entropy will do, and this is a point which can be resolved by a practical measurement, which is carried out in the forthcoming section.

The entropy rate is a property of the message which, in the context of this thesis, is hand generated material, being produced at a certain speed on a sheet of paper. The entropy rate is the average minimum number of bits per second which would be needed in theory to transmit the message so that it could be reproduced at the receiver at the same rate as it was generated at the transmitter. This is an ideal minimum bit rate.

From chapter 2, we note that the recording of our data was highly redundant. In this chapter, we are concerned with determining the entropy rate of the signal from a statistical analysis of the data which is a redundant coding of the signal. Knowing the entropy rate, one could evaluate the efficiency of any proposed coding and knowing the entropy rate is likely to help in devising an optimum coding strategy, which would include

redundancy based on an appropriate model of the transmission channel. Measuring the actual entropy rate would take years of CPU time because the computational problem grows exponentially, i.e if the alphabet of our information source, is of size S, the n'th approximation of the entropy involves S^{n+1} conditional and joint probabilities. One can make first, second, third....nth...order estimates of the entropy rate. All these estimates are greater than the actual entropy rate. Let us call it the ultimate entropy H_∞ say. Then if the nth order entropy rate is H_n, we should have $H_n \geq H_\infty$; $H_n \geq H_{n+1}$ for all n. The equal sign would hold for large n when all the correlations inherent in the data have been taken into account. For example, if n is greater or equal to the number of points in the data then $H_n = H_\infty$, because all the available information has been taken into account.

With reference to $\Delta x(t)$ and $\Delta y(t)$ correlation coefficients (Table 3.1 and figures Fig.3.1.b, Fig.3.2.b), although the autocorrelation function gradually falls off, the coefficients change slowly for a correlation distance n = 12. This suggests H_n tends asymptotically to H_∞ for n > 12. If H_n is plotted against n, a curve of the general form of Fig.3.3 should be obtained. An analytically defined curve of the following form, may be used to model the trend portrayed in Fig. 3.3.

$$H_n = Ce^{-Bn} \text{ for } n <= 12 \text{ with } B > 0 \text{ and } C > 0$$

$$H_n = H_\infty \text{ for } n > 12$$

Each computer analysis producing an estimate of H_∞ which depends on the value of n used. In doing computations, we are trying to reduce an estimate

of the entropy rate. This is done by obtaining higher order estimates of H_∞ knowing that the higher the order, the lower the estimate, but knowing that all estimates are greater than H_∞ the true entropy rate of the signal.

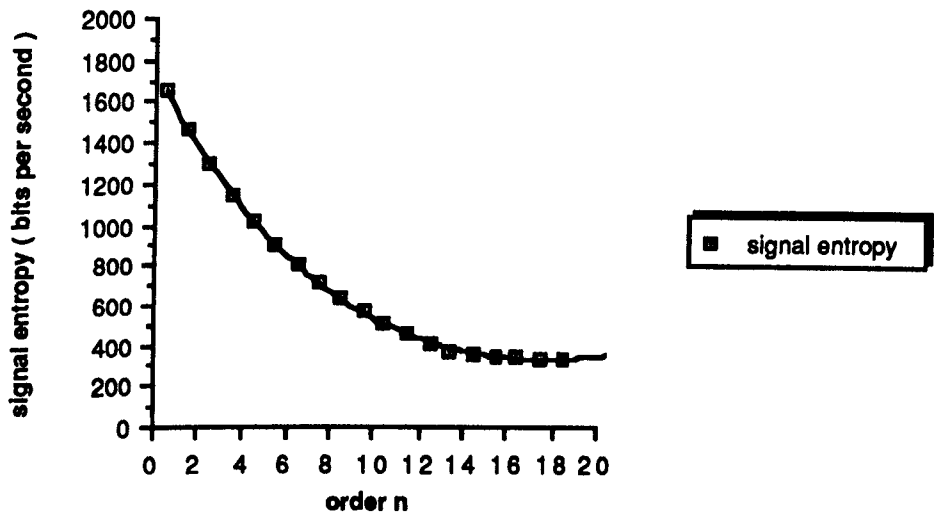


Fig.3.3 General trend of a signal entropy rate

Having briefly recalled the expressions for information measure, in section 3.2, we shall give a critical answer to each of the following questions:

1. What is the average information allocated to the first pen down position immediately after the pen up condition ?
2. Given the first order entropy per pen movement (Δx, Δy), what is the expression of the entropy of the signal associated with hand generated material ?
3. In general, when up to n'th order entropy estimates per pen movement (Δx,Δy) are known, what is the expression of the entropy of the signal associated with hand generated material ?

3.3 Measurements of the entropy estimates

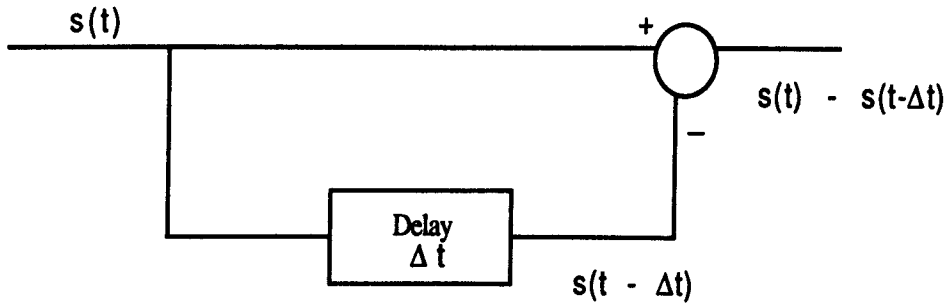
Hand drawn curves produced by a stylus and tablet combination are specified by a sufficient number of closely spaced points, or increments, or short vectors. To compute the various estimates of the entropy, we operate on coordinates captured every two hundredth of a second. Pen down counts are expanded, this implies successive replicas of identical samples, as discussed in chapter 2.

3.3.1. Technique of Measurement

The amplitude probability distribution of a stationary time series, i.e a sequence of numbers or pulses is measured by means of counting the number of occurrences of each possible value of the variable for a long time and normalizing by dividing by the total number of pulses or numbers. The hand produced data are typical time series, they were acquired from continuous handwriting signals, sampled at the highest possible rate characterizing the tablet; this was done in order to include all the significant fluctuations.

Consecutive pair of coordinates can be expressed as

$(x(t), y(t)), (x(t) + \Delta x(t), y(t) + \Delta y(t))$ to yield a set of difference vectors $(\Delta x(t), \Delta y(t))$, this a new time series on which we can apply our counting technique. The simple differencing method is illustrated in the following diagram.



The data s(t) are either x(t) or y(t)

The probabilities of various classes of the vectors, make up the distribution of our new time series. The quantification of the probability of a class of vectors is explained next.

3.3.2 Probability model between pairs of points and differences in coordinates

To simplify things let us consider first of all one dimensional analysis; let $p(\Delta x)$ be the probability of a particular class of Δx, this occurs whenever $x(t + \Delta t) - x(t) = \Delta x$; Δt is the sampling period.

Suppose the digitizing tablet gives out coordinates in the range $0 \rightarrow 2^b - 1$, then we have b bits resolution for abscissa x and b bits for ordinate y. Now x(t) can take any value from 0 to $2^b - 1$; e.g, with reference to chapter 2, the coordinates produced by PDMAKE have 9 bits resolution; thus x(t) is the range 0->511; we have Δx whenever $x(t) = k$ is followed by $x(t + \Delta t) = k + \Delta x$, thus

$$p(\Delta x) = \sum_{k=0}^{k=2^b-1} p(x(t) = k , x(t + \Delta t) = k + \Delta x) \qquad (3.5)$$

for any value of t; this expression is independent of time t for an ergodic process, that is the individual vectors are chosen from an ensemble of samples whose probabilities of occurrence do not vary with time.

$p(x(t) = k , x(t + \Delta t) = k + \Delta x)$ is the number of times $x(t) = k$ was followed by $x(t + \Delta t) = k + \Delta x$ divided by the total number of samples.

In two dimensions

$p(x(t) = k, y(t) = l, x(t + \Delta t) = k + \Delta x, y(t + \Delta t) = l + \Delta y)$ is the number of times $x(t) = k$ is followed by $x(t + \Delta t) = k + \Delta x$ when $y(t)$ and is followed by $y(t + \Delta t) = l + \Delta y$ divided by the total number of samples; so

$$p(\Delta x, \Delta y) = \sum_{k} \sum_{l} p(x(t) = k, y(t) = l, x(t + \Delta t) = k + \Delta x, y(t + \Delta t) = l + \Delta y)$$

with k and l taking values from 0 to $2^b - 1$. (3.6)

This same procedure can be used when n + 1 steps are considered, to establish the relationship between the probability of n differences and that of the n + 1 successive points.

The statistical properties of the quantized difference signals were determined by generating frequency distributions for the individual samples from each tutorial recording. As there were 13 tutorials, 13 histograms were then merged into a single frequency distribution of the required difference signal associated with the entire database. The frequency distributions of Δx, Δy were obtained by applying equations (3.5), (3.6). The probabilities are expressed with 7 significant figures after the decimal point because the total number of samples is 1283975, which is is between 10^6 and 10^7. Tables 3.2, and 3.3 describe respectively the experimental results of Δx distribution and Δy distribution. Those results are plotted in Figures 3.4 and 3.5.

Discussions of Δx, Δy distributions :

The distribution depicted in Fig.3.4 is for Δx, i.e difference between consecutive values of the horizontal signal x(t). The curve in Fig.3.5 is for Δy, i.e difference between consecutive values of the vertical signal y(t).

A study of Tables 3.2 and 3.3 shows that there are 14 classes of Δx and 10 classes of Δy with finite probabilities. Δx varies from -6 to +7 with increment of 1; the most probable Δx are -1, 0 and 1 with respective probabilities 0.0543430; 0.7967070 and 0.1359550 Δy varies from -5 to +4 with increment of 1; the most probable Δy are -1, 0 and 1 with respective probabilities 0.1160304; 0.8047887 and 0.0724991. Tables 3.2 and 3.3 show that the smaller differences (i.e absolute values) occur more frequently than the large ones. The positive Δx's represent 14.60 % of the total number of Δx classes; whereas only 5.73 % of Δx are negative, 79.67 % of Δx are 0. 7.49 % of Δy are positive, 12.04 % are negative and 80.47 % are 0. The likely reasons for a large number of $\Delta x = 0$, $\Delta y = 0$ are as follows:

1. When the movement of pen is less than the minimum resolvable element (e.g ∂) of the digitizing tablet. If $\partial = 1$, which is indeed, in our particular application, then $\Delta x = 0$, $\Delta y = 0$ will occur.
2. When the tip of the pen is dead stationary on the writing surface we can definitely expect $\Delta x = 0$, $\Delta y = 0$ to occur.
3. If the position of the pen is sampled at 200 times per second, and is moving at a speed of well below 200 quantum intervals per second, then one would expect a majority of $\Delta x = 0$ and / or $\Delta y = 0$. This will happen even in the absence of reasons 1 and 2.
   To illustrate our point, let $\Delta x = 1$, this corresponds to L cm, so at 200 samples per second a minimum pen speed of 200L cm per second would be needed for $|\Delta x| > 0$ on consecutive samples. Should the pen move at less than 200L cm per second, $\Delta x = 0$ would occur. As SMOL81 quotes 0.5 mm for the least resolvable pen move (i.e 1 unit) on the Bit pad, 10 cm per second would be needed for $|\Delta x| > 0$ on consecutive samples.

As expected most people (i.e using Roman character based languages) write from left to right; this is verified by the above results, where 14.6 % of Δx are positive, compared with 5.73 % negative.

A visual inspection of Figures 3.4 and 3.5 suggests that the relative frequency for difference in x, y direction exponentially decreases as the absolute size of the difference increases. An exponential curve of the form Ae^{Bs} (where s stands for Δx or Δy) has been fitted to both Δx and Δy distributions with the requirement it should interpolate (i.e pass through) the above most probable Δ in x / y directions.

Simple algebraic manipulations lead to:

For Δx $f(s) = Ae^{Bs}$;

where $A = 0.797$, $B = 2.6863$ for $\Delta x < 0$; $B = -1.7682$ for $\Delta x > 0$

For Δy $f(s) = Ae^{Bs}$;

where $A = 0.805$, $B = 1.9372$ for $\Delta y < 0$; $B = -2.4142$ for $\Delta y > 0$

These two analytical curves called, fitpdx, fitpdy are plotted together with the distributions pdx, pdy derived from the actual data; in Figures 3.6 and 3.7. It can be seen that the deviations from the practical curves become significant for rare classes of Δx and Δy; the more probable the Δ's classes, the better the analytical fitting.

But how good is then the analytical fitting ? Considering that we are mainly interested in the entropy of the signal and that the quantity in question is a function of the signal distributions, the question of the goodness of fit can only be fully answered when the entropy estimates from empirical and analytical curves are compared. This is attacked in the next paragraphs which deal with the analysis of the entropy estimates.

Δx	Frequency	Relative frequency
-7	0	0.0000000
-6	9	0.0000071
-5	26	0.0000205
-4	118	0.0000930
-3	421	0.0003320
-2	2291	0.0018071
-1	68895	0.0543430
0	1010049	0.7967067
1	172361	0.1359550
2	10688	0.0084305
3	2118	0.0016706
4	552	0.0004354
5	199	0.0001570
6	46	0.0000363
7	7	0.0000055

Table 3.2 Δx frequency and relative frequency distribution

Δy	Frequency	Relative frequency
-7	0	0.0000000
-6	0	0.0000000
-5	6	0.0000047
-4	54	0.0000426
-3	451	0.0003557
-2	5304	0.0041837
-1	147101	0.1160303
0	1020295	0.8047887
1	91913	0.0724991
2	2454	0.0019356
3	186	0.0001467
4	16	0.0000126
5	0	0.0000000
6	0	0.0000000
7	0	0.0000000

Table 3.3 Δy frequency and relative frequency distribution

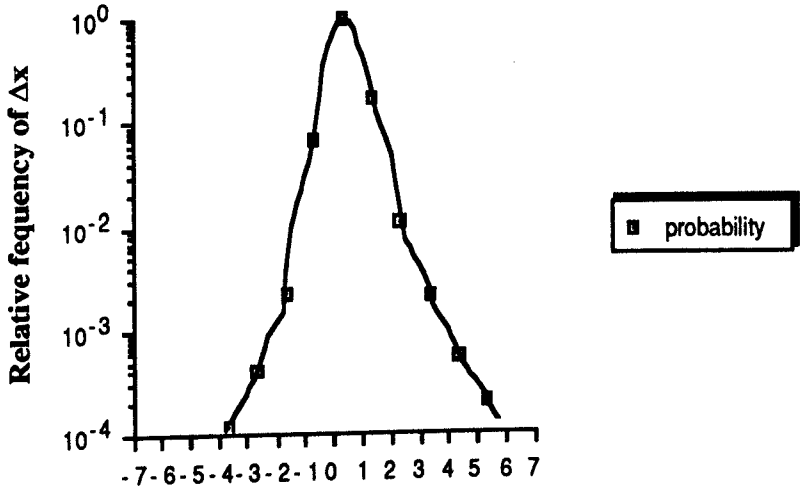


Fig.3.4 Difference signal (Δx)

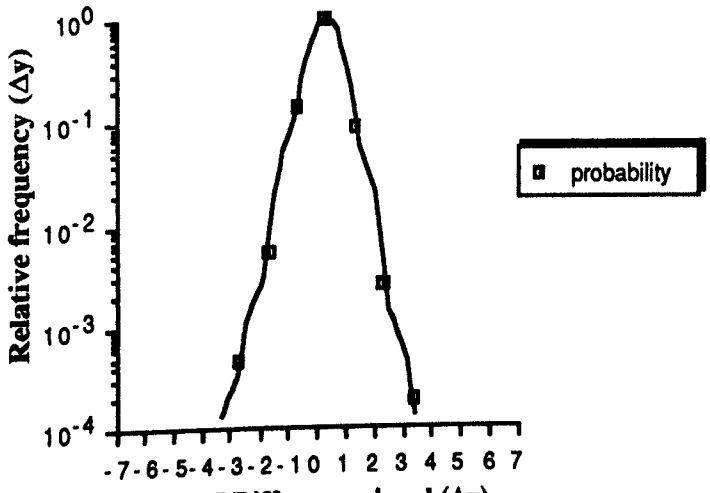


Fig.3.5 Difference signal (Δy)

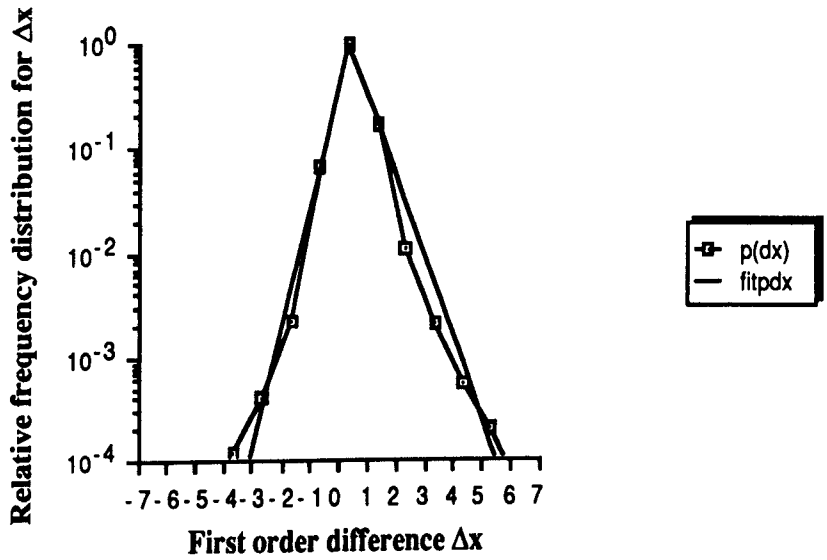


Fig.3.6. Empirical distributions (p(dx)) derived from data, and analytical distributions (fitpdx) are plotted on the same graph for visual comparison.

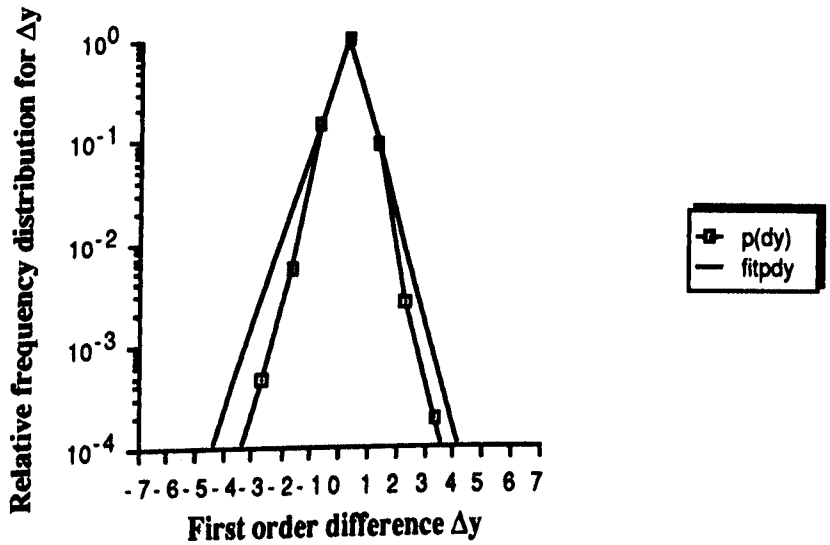


Fig.3.7. Empirical distributions (p(dy)) derived from data, and analytical distributions (fitpdy) are plotted on the same graph for visual comparison.

3.3.3 Derivation of an Entropy formula of the signal associated with handdrawn material.

As explained in section 3.2.2, the distributions used for entropy estimates were made from pulling together the histograms of 13 tutorials into one single histogram from which the relative frequency distribution was derived.

Each data file is made of data describing the traces of the pen (see chapter 2 for details). The resolution of the data dictates the number of bits allocated to the starting point of the trace, an efficient representation requires the subsequent points to be allocated a number of bits produced by equations (3.3) and (3.4). Considering these equations, if H_0 bits represent the first point of a trace, then H_1, H_2, H_3, ... H_{n-1}, H_n bits respectively represent the second, third, fourth, ..., n'th and (n+1)'th point of the trace; so for a pen trajectory made of P_s points, the total information associated with the trace is:

1. $H_0 + (P_s - 1)H_1$ when the first order entropy estimate of $(\Delta x, \Delta y)$ is known; and $P_s > 1$.
2. $H_0 + H_1 + (P_s - 2)H_2$ when the 1st and 2nd order entropy estimates are known; and $P_s > 2$.
3. $H_0 + H_1 + H_2 + H_3 + ...+ H_i + ...+ (P_s - n)H_n$ when up to nth order entropy estimates per pen movement $(\Delta x, \Delta y)$ are known; with $i < n < P_s$.

The above expressions apply only to one trace, for a database containing S traces, the general equation for $n \leq P_S$ is:

$$I_S = \sum_{s=1}^{s=S} \left(\sum_{r=0}^{r=n-1} H_r + (P_s - n)H_n \right) \quad \text{bits} \qquad (3.5)$$

Assuming, maximum order for conditional entropy is 2, the above general equation (for a hand generated material made of S = 4 traces) is interpreted as follows:

$H_0 + H_1$ for a trace consisting of 2 points.

$H_0 + H_1 + 2H_2$ for a trace consisting of 4 points.

$H_0 + H_1 + 3H_2$ for a trace consisting of 5 points.

$H_0 + H1 + rH_2$ for a trace consisting of r + 2 points.

H_0 for a trace consisting of one single point.

For this particular example where S = 4, summing up the above leads to a total entropy of $I_4 = 4H_0 + 3H_1 + 5H_2$ bits. As we have 12 points, the average information per point is I_4 / 12 bits. In general, going to a maximum order of n for the conditional entropy, we get

$I_{BP} = I_S / T_{np}$ bits per point,where $T_{np} = \Sigma P_s$

(s varies from 1 to S) is the total number of points. By (3.5), we obtain

$$I_{BP} = \left(H_0 + S \left(\sum_{r=1}^{r=n-1} H_r + \left(\sum_{s=1}^{s=S} P_s - n \right) H_n \right) \right) / T_{np} \text{ bits per point} \qquad (3.6)$$

Dividing the denominator and the numerator by the number S of traces, we get

$$I_{BP} = \left(H_0 + \sum_{r=1}^{r=n-1} H_r + (P_{av} - n)H_n \right) / P_{av} \text{ bits per point} \qquad (3.7)$$

where $P_{av} = T_{np} / S$ is the average trace length

The estimated entropy rate, i.e the average information content of the signal per unit time is I_S / D; D (in seconds) is the duration of the signal.

Equations (3.5), (3.6), (3.7) cannot be used unless nth order entropy estimates of difference signal are numerically established; this is discussed next.

3.3.4 Discussions of the results of entropy estimates of difference signals ($\Delta x(t)$, $\Delta y(t)$).

Since the signal consists of a sequence of events, the entropy estimates of interest are the conditional entropies which specify the measure of uncertainty attached to the most recent element when certain components of the signal past are known. Up to eighth order entropy estimates for difference signals have been measured and are shown in Table 3.4.

	Entropy estimates of difference signals Δx / Δy			
order	HΔx (bits)	HΔy (bits)	Total (bits)	Entropy rate (bits per seconds)
0	4	4	8	1600
1	1.020	0.900	1.920	384
2	0.892	0.889	1.781	356
3	0.800	0.800	1.600	320
4	0.736	0.736	1.472	294
5	0.702	0.702	1.404	280
6	0.686	0.686	1.372	274
7	0.678	0.678	1.356	271
8	0.670	0.671	1.341	268

Table 3.4 Theoretical entropy rate estimates (in bits per second)

The zero'th order approximation of the entropy rate assumed that all 14 classes of Δx and 10 classes of Δy, were equiprobable, providing $\log_2 14$ bits per Δx and $\log_2 10$ bits per Δy.

Looking at columns two, three and four of Table 3.4, the contribution of entropy estimates in horizontal direction (see column two) to the total entropy estimate (column four) lies between 50% to 53.1%; and for vertical direction the contribution varies between 46.9% to 50% of the total entropy estimate. The largest contribution is attributed to the horizontal direction (i.e x) and occurs at first order of entropy measurement; this means that the first order correlation coefficient is higher for y(t) signal than for x(t) signal, this result does not appreciably conflict with our previous findings on correlation analysis (section 3.2.1).

The first order entropy estimates from measured distributions, are 1.020 bits for $\Delta x(t)$ and 0.900 bits for $\Delta y(t)$. These entropy estimates are of course lower than those which may be derived from the distributions of full amplitude signal, because their distributions are less uniform. The entropy estimates from first order fitted distributions curves discussed in section 3.3.2, are 1.037 bits for $\Delta x(t)$ and 0.917 bits for $\Delta y(t)$; the differences between these and those estimated from empirical distributions are not very significant, and are respectively 1.66 % and 1.88 %. These results added to the ones in section 3.3.2 show that the fitted distribution curves are good enough.

The theoretical bit rates are shown in the last column of Table 3.4, there is an average of 7.56 % appreciable decrease in entropy rate from first order to fifth order of entropy measurement, but from sixth order the average decrease in entropy rate estimates drops to 1.16 %. These results suggest that the higher the order in entropy rate, the lower the correlation between

successive samples; this trend agrees with what is expected.

The trace length statistics have been measured, and are shown in Table 3.5 Fig.3.8 depicts the frequency distribution of number of counts per segment (including equal coordinates) against number of times each count occurs. Fig.3.9 depicts the cumulative relative frequency distribution of number of counts per segment.

An analysis of Table 3.5, Figures 3.8 and 3.9 showed that :

a. The shortest segment length was 1 sample.
b. The longest segment length was 2768 samples.
c. The segment length of 24 samples, was the most frequent.
d. The average segment length was 80 samples.
e. The total number of segment lengths was 14557.
f. 70 % of segment lengths were less than 80 samples.
g. 90 % of segment lengths were less than 157 samples.
h. 95 % of segment lengths were less than 250 samples.
i. The probability of segment length greater than 226 samples, varies between 0.00007 and 0.00021.
j. The probability of segment length less than 10 samples, varies between 0.00007 and 0.0056.

Length	Frequency	Length	Frequency	Length	Frequency	Length	Frequency
1	9	2	1	5	3	6	21
7	48	8	42	9	80	10	79
11	114	12	117	13	156	14	198
15	181	16	193	17	211	18	196
19	197	20	231	21	209	22	238
23	207	24	245	25	212	26	220
27	177	28	211	29	210	30	181
31	157	32	172	33	160	34	147
35	167	36	181	37	148	38	165
39	161	40	162	41	141	42	164
43	154	44	176	45	156	46	141
47	140	48	132	49	144	50	121
51	128	52	152	53	114	54	128
55	123	56	125	57	149	58	125
59	118	60	110	61	140	62	94
63	91	64	106	65	102	66	98
67	78	68	75	69	71	70	82
71	95	72	76	73	58	74	69
75	70	76	73	77	77	78	84
79	73	80	56	81	55	82	58
83	64	84	81	85	56	86	62
87	59	88	64	89	73	90	65
91	59	92	58	93	64	94	39
95	61	96	53	97	56	98	55
99	67	100	56	101	57	102	49
103	34	104	50	105	56	106	54
107	51	108	42	109	36	110	35
111	37	112	36	113	35	114	28
115	36	116	33	117	38	118	29
119	28	120	31	121	46	122	27
123	32	124	29	125	25	126	34
127	40	128	22	129	23	130	22
131	26	132	17	133	30	134	34
135	26	136	25	137	25	138	22
139	23	140	21	141	26	142	23
143	27	144	20	145	31	146	26
147	13	148	18	149	14	150	13
151	23	152	13	153	22	154	14
155	21	156	12	157	11	158	17
159	10	160	12	161	23	162	17
163	14	164	11	165	19	166	4
167	6	168	16	169	11	170	14
171	17	172	15	173	19	174	18
175	14	176	10	177	7	178	12
179	6	180	12	181	11	182	10
183	7	184	10	185	13	186	8
187	14	188	11	189	11	190	8
191	15	192	9	193	6	194	9
195	10	196	9	197	5	198	7
199	4	200	7	201	14	202	10
203	10	204	7	205	3	206	8
207	7	208	8	209	7	210	2
211	7	212	8	213	3	214	5
215	6	216	9	217	6	218	4
219	4	220	4	221	4	222	11

Table 3.5. Statistics of pen runs.

Length	Frequency	Length	Frequency	Length	Frequency	Length	Frequency
223	5	224	7	225	5	226	1
223	5	224	7	225	5	226	1
227	7	228	5	229	4	230	3
231	6	232	8	233	10	234	7
235	5	236	4	237	6	238	4
239	1	240	7	241	5	242	8
243	3	244	8	245	2	246	4
247	6	248	4	249	3	250	8
251	8	252	4	253	3	254	3
255	3	256	7	257	8	258	3
259	7	260	7	261	1	262	7
263	4	264	4	265	1	266	4
267	4	268	4	269	7	270	5
271	8	272	4	273	3	274	6
275	3	276	1	277	3	278	4
279	6	280	1	281	3	282	7
283	2	284	2	285	1	286	4
287	5	289	1	290	2	291	5
292	5	293	1	294	4	295	3
296	1	297	2	298	1	299	5
300	4	301	7	302	3	303	4
304	2	306	5	307	4	308	1
309	4	310	4	311	5	312	3
313	1	314	2	315	4	316	2
317	1	318	4	319	4	320	7
321	5	322	4	323	4	325	3
327	4	328	1	329	4	330	5
332	1	333	4	334	3	335	3
336	1	338	7	340	9	341	2
342	5	343	5	344	3	345	1
346	2	347	1	348	1	349	3
350	3	351	1	352	3	353	1
354	1	355	7	356	1	358	1
359	1	361	1	362	3	363	1
364	1	365	5	366	1	367	5
368	1	370	2	373	2	374	6
375	1	376	4	377	2	378	2
379	3	380	2	381	2	382	1
383	1	385	1	386	2	388	3
389	2	391	1	392	1	394	4
395	2	397	1	398	4	399	3
401	1	402	1	403	1	404	2
405	3	406	1	407	2	408	2
409	4	410	4	411	1	412	1
414	3	415	4	417	1	418	1
419	1	420	2	421	1	423	2
425	1	428	2	429	1	430	1
431	2	432	1	433	2	434	2
437	1	438	2	439	1	440	1
441	1	442	1	443	2	444	2
445	1	446	3	448	1	449	3
452	1	453	1	454	1	456	2
462	2	463	1	464	1	465	2
466	2	468	2	470	3	472	4
474	3	480	1	481	3	483	1

(Continued from Table 3.5)

Length	Frequency	Length	Frequency	Length	Frequency	Length	Frequency
484	1	485	1	486	2	491	1
493	1	494	2	495	1	496	1
497	1	498	2	499	1	501	1
503	1	505	1	509	1	512	2
513	1	515	1	516	2	517	1
518	1	520	3	522	1	525	1
529	2	530	1	531	1	533	2
535	1	539	1	541	1	544	1
545	1	547	1	548	1	549	1
553	1	556	2	561	1	562	1
566	1	568	1	570	2	572	1
577	1	579	1	580	1	583	4
584	1	585	1	586	1	587	1
588	1	593	1	594	1	595	2
596	1	600	2	607	1	608	1
609	2	612	1	615	1	617	2
619	1	621	1	622	3	626	1
640	2	646	1	655	1	660	1
664	1	667	1	670	1	677	1
678	1	680	1	683	1	685	1
690	1	696	1	706	1	710	1
712	1	716	1	717	2	718	1
719	1	721	1	724	1	726	1
733	1	734	1	759	1	760	1
771	1	781	1	782	1	783	1
806	1	812	1	824	1	825	1
837	2	851	1	853	1	859	1
873	2	883	1	899	1	910	1
929	1	930	1	966	1	972	1
977	1	982	1	1012	1	1028	1
1032	1	1055	1	1094	1	1108	1
1117	1	1134	1	1154	1	1158	1
1224	1	1233	1	1255	1	1269	1
1287	1	1324	1	1369	1	1407	1
1486	1	1541	1	1565	1	1570	1
2581	1	2768	1				

(continued from Table 3.5)

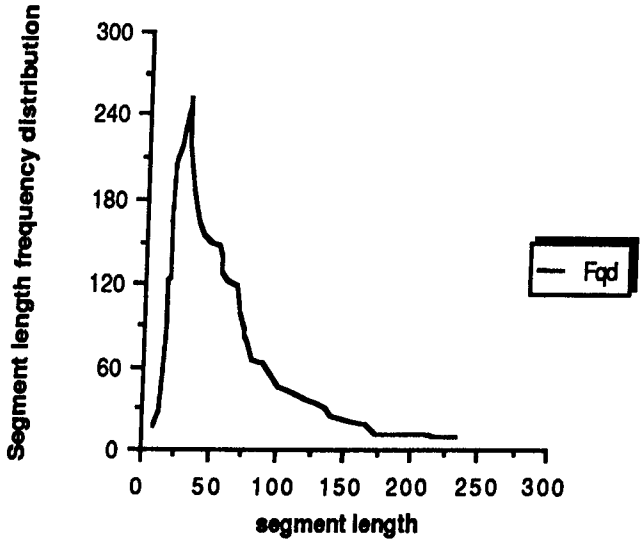


Fig.3.8 Distribution of segment lengths

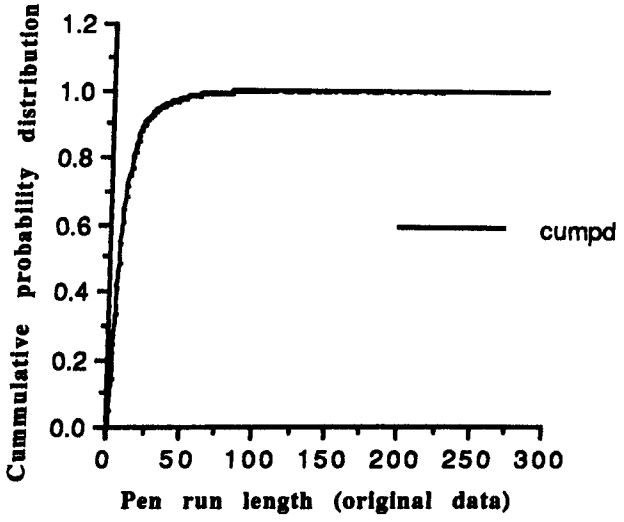


Fig.3.9 Cumulative relative frequency distribution of segment length.

	Theoretical and practical entropy rate estimates (in bits per second)	
order	Theoretical estimate	practical estimate
0	1600	1625
1	384	424
2	356	397
3	320	362
4	294	338
5	280	325
6	274	319
7	271	316
8	268	314

Table 3.6 Theoretical and practical entropy rate estimates

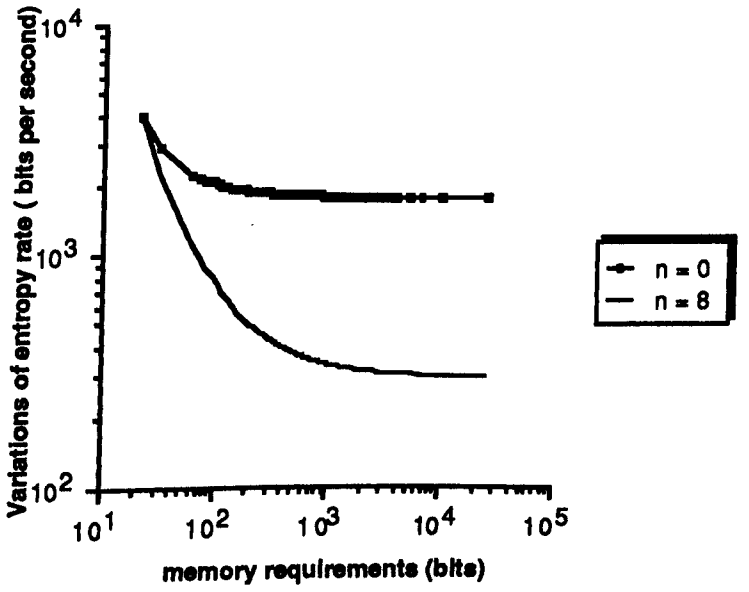


Fig.3.10 Bit requirements versus entropy rate

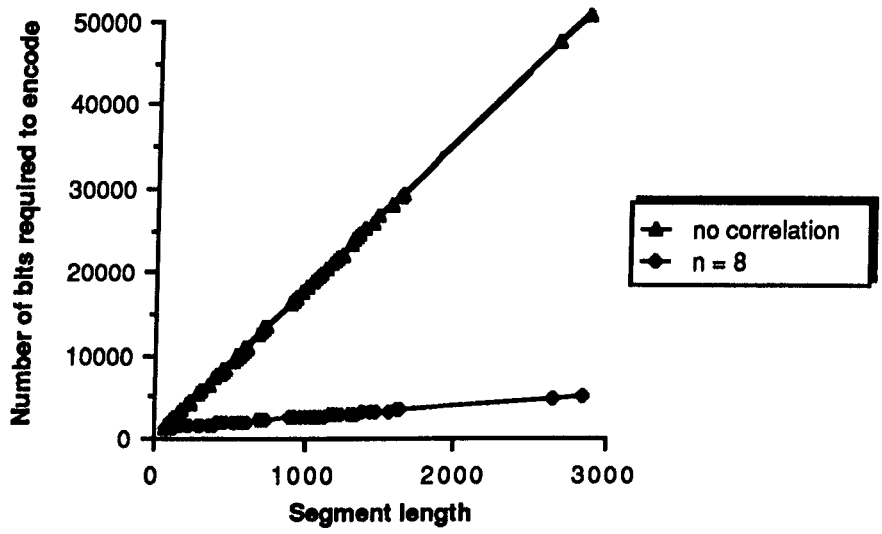


Fig.3.11 Bit requirements versus segment length

Having got the estimates of entropy (column 4 of Table 3.4), equation (3.5) and the statistics of the lengths of the traces enable us to determine the practical limits of entropy rates.

The theoretical estimates and practical limits of entropy rates are displayed in Table 3.6. The variations of entropy rate versus the bit requirements to encode "pen trace segment", are shown in Fig.3.10. Two curves are displayed in Fig.3.10.

The curve labelled (n = 0) is associated with the zero order correlation. In this case, it was assumed that Δx and Δy have flat distributions.

The curve labelled (n = 8) is associated with 0 up to eighth order correlation between successive pen movements (i.e Δx, Δy).

For n = 0 the lowest possible entropy rate was about 1600 bits per second.

For n = 8 the lowest possible entropy rate was about 269 bits per second.

We can see that for both curves, the entropy rate is a decreasing function of the bit requirement to encode "pen trace segment"; i.e the lower the entropy rate, the higher the number of bits required.

The bit requirements versus segment length are shown in Fig.3.11. Here two curves are shown. We can see that they are straight lines, so there is a linear relationship between the segment length and the bit requirement to encode. When no correlation is assumed, the demand in storage requirement is higher than when up to eighth (n = 8) order correlations are assumed. Straight lines were fitted to the (bit requirement, segment length) data and were found to be:

$B_r = 18\ S_L$; when no correlation was assumed (n = 0).

$B_r = 1.34\ S_L + 17.58$; when up to 8th order correlations were assumed (n = 8).

Here B_r stands for bit requirement; and S_L for segment length.

The highest practical limit of entropy rate (order of correlation n = 0) is obtained, when the classes of Δx, Δy are assumed equiprobable.
When we compare the practical limits of entropy rates, with the target bit rate of 200 bits per second or less, it is clear that the lowest practical limit of entropy rate is higher by 54.91 %.
This result is unacceptable and unexpected, because intuitively one should expect a substantially lower zero order entropy rate and subsequent substantial rates of decrease for higher order, when a signal is highly correlated; in our case the signal is highly correlated because it was sampled at the highest possible rate provided by the manufacturer of the Bit Pad.

Δx	Δy	Frequency	Relative frequency	Δx	Δy	Frequency	Relative frequency
-6	-1	1	0.0000008	1	-5	2	0.0000015
-6	0	7	0.0000055	1	-4	9	0.0000070
-6	1	1	0.0000008	1	-3	81	0.0000638
-5	-1	3	0.0000023	1	-2	719	0.0005671
-5	0	23	0.0000181	1	-1	16512	0.0130243
-4	-1	24	0.0000189	1	0	135801	0.1071171
-4	0	78	0.0000615	1	1	18746	0.0147864
-4	1	16	0.0000126	1	2	461	0.0003636
-3	-2	11	0.0000086	1	3	27	0.0000213
-3	-1	105	0.0000828	1	4	3	0.0000023
-3	0	261	0.0002058	2	-5	2	0.0000015
-3	1	42	0.0000331	2	-4	17	0.0000134
-3	2	2	0.0000015	2	-3	8	0.0000063
-2	-3	7	0.0000055	2	-2	49	0.0000386
-2	-2	85	0.0000670	2	-1	538	0.0004243
-2	-1	493	0.0003888	2	0	8379	0.0066092
-2	0	1541	0.0012155	2	1	1597	0.0012596
-2	1	158	0.0001246	2	2	84	0.0000662
-2	2	6	0.0000047	2	3	13	0.0000102
-2	4	1	0.0000008	2	4	1	0.0000008
-1	-5	1	0.0000008	3	-3	1	0.0000008
-1	-4	9	0.0000070	3	-2	4	0.0000031
-1	-3	60	0.0000473	3	-1	58	0.0000457
-1	-2	629	0.0004961	3	0	1635	0.0012896
-1	-1	13331	0.0105152	3	1	367	0.0002894
-1	0	49395	0.0389618	3	2	42	0.0000331
-1	1	5195	0.0040977	3	3	10	0.0000078
-1	2	248	0.0001956	3	4	1	0.0000008
-1	3	26	0.0000205	4	-1	18	0.0000142
-1	4	1	0.0000007	4	0	396	0.0003123
0	-5	1	0.0000007	4	1	117	0.0000922
0	-4	19	0.0000150	4	2	17	0.0000134
0	-3	294	0.0002319	4	3	4	0.0000031
0	-2	3807	0.0030028	5	-1	5	0.0000039
0	-1	116013	0.0915087	5	0	158	0.0001246
0	0	822590	0.6488428	5	1	30	0.0000236
0	1	65632	0.0517692	5	2	1	0.0000008
0	2	1593	0.0012565	6	0	29	0.0000228
0	3	96	0.0000757	6	1	8	0.0000063
0	4	4	0.0000031	6	3	4	0.0000031
				6	4	5	0.0000039
				7	0	2	0.0000015
				7	1	4	0.0000031
				7	3	1	0.0000008

Table 3.7 Δx, Δy frequency distribution (x / y correlation assumed)

So, What has gone wrong ?

When x/y correlation is assumed (see Table 3.7), the zero order statistics of first difference signals show that 65 % of (Δx, Δy) are (0, 0); this is due to quantization noise, which happens when the dimension of the writing pen is smaller than the smallest resolvable element of the digitzing pad (see chapter two). Apart from the noise due to the discreteness of the Pad, noise due to the jittering of the hand must be considered. All these noise sources affect the data used to estimate the probabilities. As the entropy is a function of probabilities, we can conclude that the entropy estimates do not fall off significantly as expected, because the data are noisy. If noise is removed or at least reduced, we will obtain yet again another new time series which can be operated on, in the manner described previously. Hence new probability distributions will likely lead to more acceptable entropy rate estimates. Our point is clarified by the following analysis :

Assuming x/y correlation, let us say successive replicas of (x, y) occur because of slow writing. If we suppress successive identical (x, y) but one, this not deemed significant because, in the scope of our specific application, it does not alter the visual information conveyed by the signal.
The implication of this simple operation leads to the elimination of
($\Delta x = 0, \Delta y = 0$) from the set of pen movements.

What is the effect on first order delta entropy rate of deleting the most probable delta value ?

For the purpose of our discussions, the coordinate pairs (x, y) are transformed from two dimensional to one dimensional by the following one to one function:

Let X_{max} , Y_{max} be the maximum values of x and y in our data, then the

required function is

$$D(t) = Y_{max} * x(t) + y(t) \qquad (3.8)$$

For consecutive pairs (x_1, y_1, x_2, y_2), the function is

$$D_2(t) = K_2 * K_3 * K_4 * x_1 + K_3 * K_4 * y_1 + K_4 * x_2 + y_2 \qquad (3.9)$$

In general, for consecutive n-tuple of coordinate pairs $(x_1, y_1, x_2, y_2, \ldots\ldots x_n, y_n)$, the general one to one function is

$$D_n(t) = K_2 * K_3 .. K_{2n-1} * K_{2n} * x_1 + K_3 * K_4 .. * K_{2n-1} * K_{2n} * y_1 + \ldots\ldots + K_{2n} * x_n + y_n \qquad (3.10)$$

where $K_{2n-1} = X_{max}$, $K_{2n} = Y_{max}$

Our analysis can be applied to any consecutive n-tuple of coordinate pairs (x,y); this justifies the reason for formulae (3.9), (3.10).

With reference to (3.8), Δ will refer to $D(t + \Delta t) - D(t)$. If the most probable Δ is not retained, in practice, this will mean not retaining the $\Delta = 0$ case. Consider a signal sampled S times per second. Let the alphabet of deltas have n distinct elements each with probability $P(\Delta_i) = P_i$; with

$$P_i \geq 0$$

$$\text{and } \sum_{i=1}^{n} P_i = 1 \qquad (3.11)$$

The entropy per sample, H_{in} say, is

$$H_{in} = -\sum_{i=1}^{n} P_i \log_2 P_i \qquad (3.12)$$

and the entropy rate when no filtering is applied, is H_{in} S bits per second,

i.e $\mathbf{R_{in} = H_{in} S}$ **(3.13).**

Now let us assume that the greatest P_i is P_1 and the delta corresponding to P_1 is suppressed.

In time T we have TS samples, of these TS P_1 are removed. Hence we retain TS (1 - P_1) elements in time T and the alphabet has n - 1 elements. The ith element occurs TS P_i times in time T, so the probability Sri for the retained element is

$$\mathbf{P_{ri} = P_i / (1 - P_1)} \qquad \textbf{(3.14)}$$

The entropy for the retained signal is

$$\mathbf{H_{out} = -\sum_{i=2}^{n} P_{ri} \log_2 P_{ri}}$$

The new entropy rate is R_{out} = Hout S (1 - P1) bits per second, i.e

$$\mathbf{R_{out} = \left(-\sum_{i=2}^{n} \{ P_{ri} \log_2 P_{ri} \} \right) S(1 - P_1)}$$

Using (3.13), simple algebraic manipulations lead to

$$R_{out} = - \{ \sum_{i=1}^{n} P_i \log_2 P_i - \sum_{i=1}^{n} P_i \log_2(1 - P_1) - P_1 \log_2 P_1 + P_1 \log_2(1 - P_1) \} S$$

So, using (3.11) and (3.12)

$$R_{out} = \{ Hin + \log_2(1 - P_1) + P_1 \log_2 P_1 - P_1 \log_2(1 - P_1) \} S$$

or

$$\mathbf{R_{out} = \{ H_{in} + (1 - P1) \log_2(1 - P_1) + P_1 \log_2 P_1 \} S} \qquad \textbf{(3.15)}$$

Hence the " coding advantage " gained by removing the most probable element is , using (3.13)

$$\mathbf{R_{in} / R_{out} = H_{in} / (H_{in} + (1 - P_1) \log_2(1 - P_1) + P_1 \log_2 P_1)}$$

Where P_1 is the probability of the most probable element.

Using the probabilities in Table 3.5, H_{in} = 1.92 bits per (Δx, Δy);

P_1 = 0.65 and S = 200 samples per second produce:

Rout = $\{1.92 + 0.35\log_2 0.35 + 0.65\log_2 0.65\}*200$ =197.6 bits / s

The gain is R_{in} / R_{out} = 1.94 times.

The deletion of $\Delta = 0$ should be the result of smoothing or spatial filtering applied to the original data.

It is interesting to note that in (3.15), the terms

$(1 - P_1)\log_2(1 - P_1) + P_1\log_2 P_1$ is the negative of the entropy of a 2-state source with state probabilities P_1 and $(1-P_1)$.

Hence we know that

Max $\{ (1 - P_1)\log_2(1 - P_1) + P_1\log_2 P_1 \} = -1$.

This occurs when $P_1 = 1 / 2$. This implies that Hin >1 if $P_1 = 1 / 2$.

The variations of the gain (R_{in} / R_{out}) are shown in Fig.3.12. R_{in}/R_{out} is of the form x / (x + K). For very large input entropy (i.e x tends to $+\infty$), the gain tends towards 1. For very small input entropy (i.e x tends to 0), the gain becomes large; however K being negative, the lowest practical input entropy must be x = -K.

If H_{in} is fixed (1.92 in our experimental data), the curve of the gain (Rin / Rout) against the probability (p_1) of the most probable Δ, is shown in Fig.3.13. We can see that the highest gain is obtained is when $p_1 = 0.5$. This, then, yields the lowest entropy rate, which is below 200 bits per second as shown in Fig.3.13.

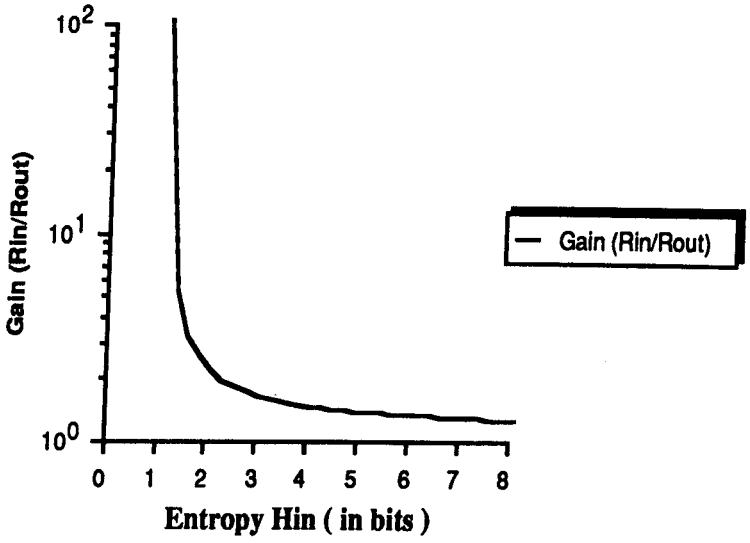


Fig.3.12.

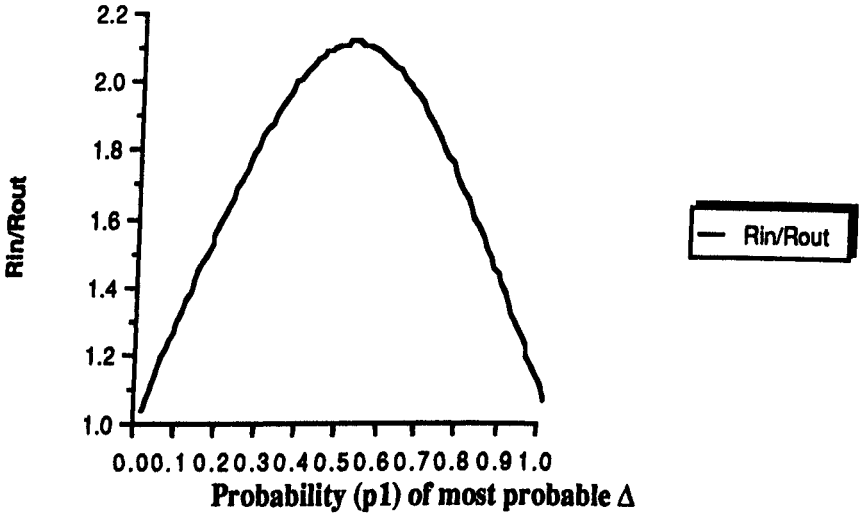


Fig.3.13

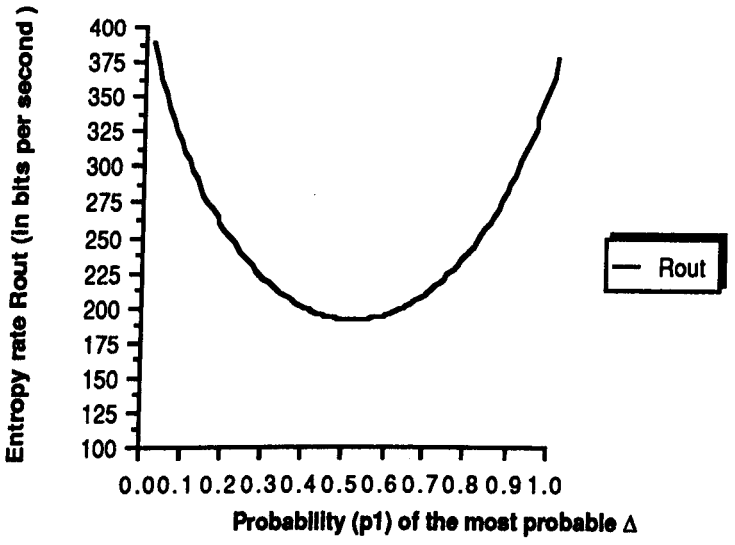


Fig.3.14

The above analysis clearly implies that , reducing pictorial information by eliminating those aspects of visual appearance of the picture that contribute little to the picture's interpretation or perception by a human observer, should result in the required entropy rate. Finding ways to do this, is the object of the next chapters.

3.4 Conclusions.

This chapter suggests that, unless the quantized handwriting and drawing signals are subject to a suitable transformation (i.e smoothing), it is not possible to code the signal so that the bit rate is at most 200 bits per second. To achieve this goal, in the forthcoming chapters, we propose to cleverly degrade the trajectory of the pen, in a such a way that the human observer would not reject it, provided that the degraded picture conveys the intended message (or information). By degrading the signal specification, the inherent redundancy will be reduced.

4. CONSIDERATIONS OF THE EFFECTS OF CONSTANT SUBSAMPLING RATE

In this chapter, we investigate the estimates of the entropy of time sub-sampled handwriting and drawing signals. The curves describing the trajectories of the pen are segmented in time, at constant rate. Bit rates are estimated on the basis of points generated by the segmentation process.

4.1 Introduction

The highest sampling rate available from the Bit Pad one (SUM79), 200 samples per second, was used, although it was felt intuitively as well as on the basis of initial tests, that this was a much higher sampling rate than necessary. Indeed various commercial systems (READ81) operate satisfactorily at 40 samples per second. At first sight, all we need to do is to sample at the Nyquist rate; i.e at a sampling frequency which is twice the highest frequency present in the signal. But this may be too high a sampling rate; at least from the point of view of picture transmission and reconstruction at the receiver if the higher frequency components of the system represent noise or contribute to a degree of resolution which is either too fine to be perceived by viewers or to be reproduced on the receiving screens. Another contribution to the higher frequencies can be due to small unwanted wobbles, jerks, or unwanted excursions of the pen that may result from erratic hand motions in positioning the pen upon the paper or in overcoming the initial friction when writing begins. Yet another contribution to the higher signal frequencies could be due to tremors of the writer's hand or to irregularities due to stiction or other similar effects as the pen moves against the paper.

We cannot just reduce the sampling rate to correspond to the highest frequencies that need to be transmitted for satisfactory reproduction of the

graphic material, because of the well known aliasing effect that would arise from any frequencies present above the highest useful frequency value. In many systems one could use an analogue anti-aliasing filter to remove the unwanted frequencies before the signal is sampled. Unfortunately, from this point of view, The Bit Pad One uses sampling as an inherent part of its signal generation and there is no continuous signal available for analogue filtering.

The solution is to sample at a rate sufficiently high to avoid aliasing and then to use a digital filter to remove the high frequency components which are not needed to reproduce an acceptable picture. One can then use a suitable process to reduce the effective sampling rate before further processing.

The provision of a maximum sampling rate of 200 samples per second, for the digitising tablet (SUM79) is presumably the result of the manufacturer practical experience that this is adequate for all uses to which the tablet is put.

The main question to be resolved in this chapter is that of the highest frequency which needs to be present in the signal in order to produce acceptable pictures from handwriting and handdrawn material. Given this knowledge and a method of filtering any components above this highest frequency, we can deduce the maximum necessary sampling rate for picture processing, transmission and reproduction.

Conversely, by using digital filters to remove high frequency components using decimation to reduce the effective sampling rate, we can produce a set of pictures whose highest frequencies have been reduced in a controlled way. The lowest effective cut off frequency to give us acceptable pictures, gives us a measure of the highest useful frequency in the signal and of the minimum sampling rate. The main part of this chapter is concerned with determining the minimum sampling rate.

Summarising, if the high frequency components in the signal produced by the digitising tablet using its maximum sampling rate correspond to noise or to picture detail of greater resolution than that which can be displayed or perceived in practice, we low pass filter the signal to remove the higher frequency components.

Displaying and inspecting the pictures produced from the filtered signal, may enable us to determine the limit of acceptability for picture impairments, and hence the highest useful frequency in the signal and the minimum required sampling rate.

4.2 Considerations of digital filters for decimation

CROCHIERE81, SCHAFER73 present a theoretical treatment of decimation filters. Decimation is a process of data reduction involving the selection of samples of the digitized data at uniformly spaced intervals throughout the data sequence, so the process of decimation involves a sampling rate reduction. The positions of the pen { P_n }, were generated every 0.005 seconds, that is , a sampling frequency F_S = 200 Hz. It is known that if one has a digital sequence {P_n }, its digital spectrum PS(f) is periodic and consists of the spectrum of the analog signal P(t) repeated infinitely around +/-k200 multiples of the sampling frequency (k = 0, 1, 2, 3 ...). Consider a sequence {P_n } derived by sampling at 200 samples per second. We wish to reduce the sampling rate to 200 / D, that is decimate with a ratio of D. Obviously, it only makes sense to reduce the sampling rate if the information content of P(t) we wish to preserve is bandlimited to less than 200 / 2D, half the desired sampling rate, since any spectral components above this frequency will be aliased into frequencies below 200 / 2D (CROCHIERE81). Fig.4.1 depicts the digital spectrum of a

typical sequence $\{P_n\}$ that we wish to decimate. The frequency band of interest is [0 - fc], and we have to ensure that, in the decimation process, no undesirable frequency components are aliased back into it. It is clear from Fig.4.1 that the first step in the decimation process has to be the filtering of the sequence $\{P_n\}$, so as to ensure that the energy left above 200 / 2D is less than a suitably chosen minimum, for example we may want to keep the aliasing error at 50 dB below the peak of Ps(f). Once the signal is appropriately filtered, it can be decimated by simply dropping the unneeded samples; for example, only every Dth sample is preserved

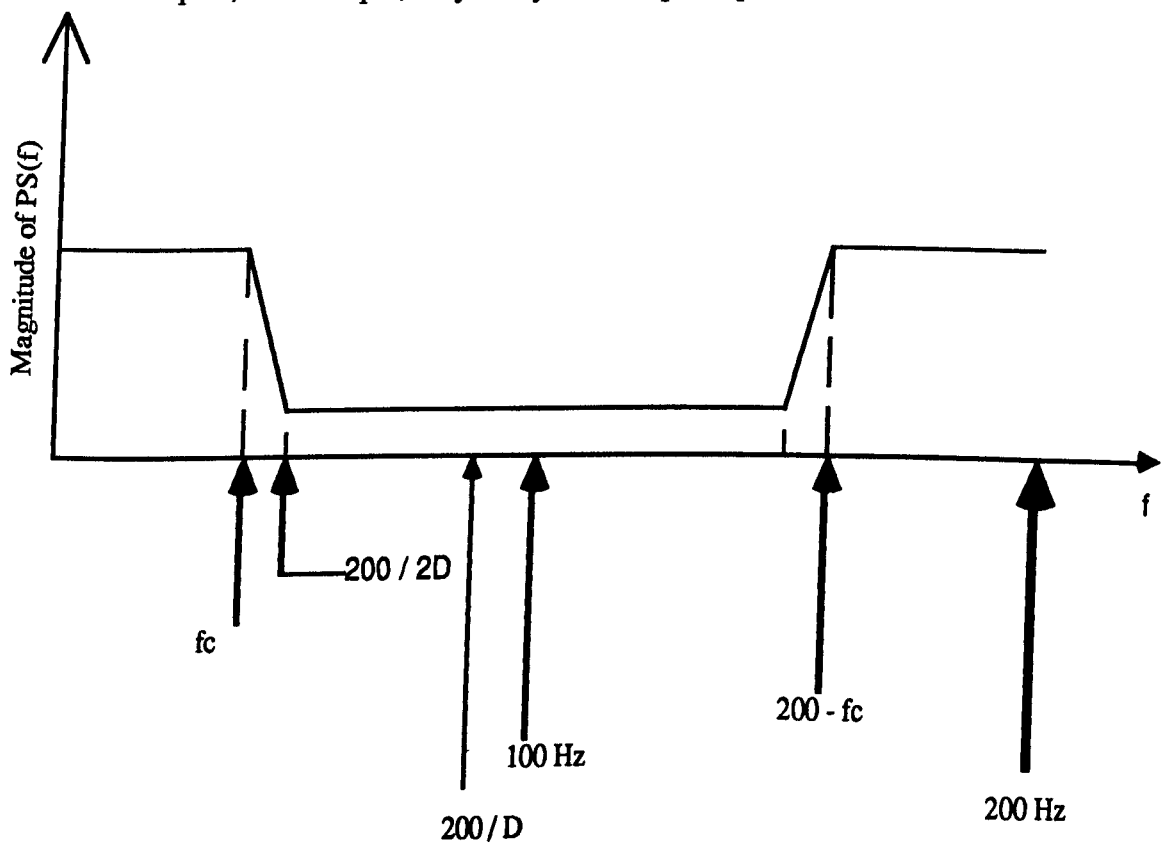


Fig.4.1 Typical spectrum of signal to be decimated

Fig.4.2 depicts the schematic diagram of a general integer ratio D decimator. We see that the signal describing the trajectory of the pen is first passed through a lowpass filter that passes undisturbed the desired information band and attenuates the band 200 / 2D to 100 Hz to prevent an excessive

aliasing error. The output of the lowpass filter passes through the decimator which simply keeps every Dth point. Now what kind of filter should we choose ?

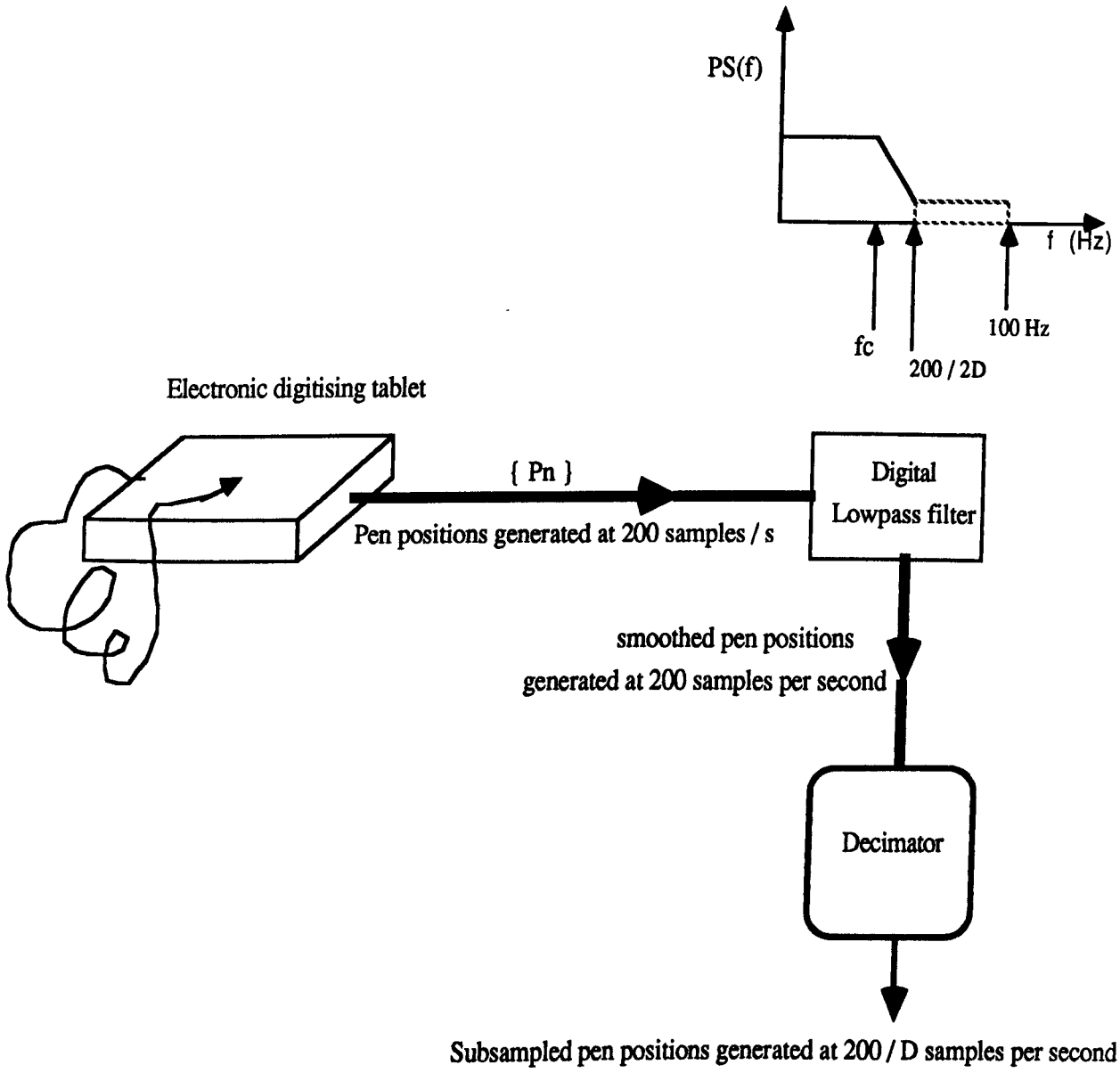


Fig.4.2. Stages involved in a decimating process

4.2.1 Choice of the filter

It can be seen that we do not have to compute the output of the lowpass filter at the rate 200 samples per second, but rather can compute only samples at the rate 200 / D.

We have taken into account this observation in deciding whether we should use an IIR (i.e recursive) or FIR (i.e non recursive filter) for the lowpass filtering. Using an IIR filter in this case has an obvious drawback. We cannot take advantage of the fact that we have to compute only every Dth output, since previous outputs are needed to compute the Dth output. Consequently no saving is achieved. However, using an FIR filter in this case implies that we can do our output computation at the rate 200 / D. Thus using an FIR filter in the decimation process leads to a significantly lower computation time. Another advantage of using an **FIR filter** is the fact that we can design **linear phase filters, and this is desirable in our application.** Usually phase does not matter for sound, but in the framework of pictorial processing, it does.

Further advantages of using FIR filters in decimators are well documented in CROCHIERE81 and references there in.

4.2.2 Digital lowpass FIR filter design

There is a great deal of literature on designing non recursive digital lowpass filter (CROCHIERE81 and references there in). Considering that filter design techniques are available and widely reported, we concentrated instead on how we should apply the filter in our particular case. Before applying filtering to our data, the following questions needed resolving:

a. How much filtering is needed to give the optimum signal to noise ratio ?

b. How will the filtering procedure affects the signal ?

Having studied various design techniques, it became clear that most of them did not address the above questions. Since the required signal properties depend on factors of human perception, there is no suitable information. The optimum S / N and effect of filtering can only be assessed by user tests on the final product.

The computational algorithm for the filtering process is assumed to be that

of a linear weighting of the original pen coordinate data values. The weighting factors are simply the coefficients of the filter. Thus the filter output data (sx_n, sy_n) are calculated from the original pen coordinates data (x_n, y_n) by the weighting :

$$sx_n = \sum_{k=-N_p}^{k=N_p} b_k \, x_{n-k}$$

$$sy_n = \sum_{k=-N_p}^{k=N_p} b_k \, y_{n-k} \qquad (4.1)$$

Where b_k are the filter weighting coefficients; the total number of terms in the filter is $2N_p + 1$; $2N_p$ is the span of the filter in samples. The output is simply the convolution of the input, with the filter coefficient sequence b_k. The filter design is then that of determining a set weighting factors b_k, giving the desired smoothing action.

A brick wall lowpass filter (Fig.4.3), would meet the requirement ideally; but this cannot be realized. So the best we could do was to adopt a filter with a frequency characteristic similar to that of the brick wall low pass filter. With this target in mind, we thought that a filter with a monotonic frequency response, maximally flat pass and stop band characteristic could suit our purpose. From KAISER66, the magnitude characteristic of such a filter is given by :

$$H(y) = (1 - y)^K \sum_{n=0}^{L-1} ((k - 1 + n)! / (K - 1)!n!) \, y^n \qquad (4.2)$$

where $y = (1 - \cos(2\pi f / f_s) / 2$ and

K is the order of tangency at $f = f_s / 2$
L is the order of tangency at $f = 0$ and

f_S is the sampling frequency in Hz.

So this maximally flat pass and stop band design technique is characterized by a tangency of the filter magnitude frequency characteristic of Lth order at zero frequency and Kth order at half sampling frequency. The combined order is 2(K + L - 1).

Equation (4.2) is rewritten in a form more convenient for calculation as :

$$H(y) = (1 - y)^K \left(1 + \sum_{n=1}^{N_T - K} \left(\prod_{i=1}^{K-1} (1 + n/i) \right) y^n \right) \quad (4.3)$$

where $N_T = K + L - 1$, is the half order of the filter

To find the actual coefficients of the filter having the magnitude characteristic given in equation (4.2) with the expression given y, it is necessary to rewrite H(y) in the form

$$H(f) = b_1 + \sum_{n=1}^{N_T} 2\, b_{n+1} \cos(2\pi n f / f_S) \quad (4.4)$$

The resulting filter coefficients can then be used directly on the data, the filtered data is calculated as expressed in equation (4.1). The design equation has been approximated roughly by (KAISER74):

$$N - 1 = 2 * N_T = 1 / 2\beta^2 \quad (4.5).$$

where N -1 is the order of the filter and ß the normalized transition bandwidth as measured by the region where the filter response magnitude varies from 95 % (passband edge) down to 5 % (stop band edge). It can be noted that the number of terms required is inversely proportional to the square of the transition width. For very small values of ß, we can expect a great deal of terms to deal with. Usually, the design proceeds as follows:

Stage a. Specify the normalized cut off frequency f_c, and the normalized transition width ß. The normalization is with respect to the sampling frequency; i.e $0 < f_c < 0.5$, $0 <$ ß $<$ minimum $(2f_c, 1 - 2f_c)$.

A suitable design should require f_c and ß to be inside these limits.

Using the approximate design formula (4.5), we can determine a lower estimate of the required filter order, N - 1

Stage b. the degree of tangency at zero frequency is then determined such as to give the desired location for the cutoff frequency, f_c , of the transition region. The filter order is permitted to change by up to a factor of 2 over that determined in stage a; in order to adjust the cutoff frequency f_c, to as close to the prescribed value as possible.

Stage c. The filter coefficients are then determined by first evaluating the frequency response characteristic at an equally spaced set of frequency points and then performing an inverse discrete Fourier transform on this set of points. Advantage is taken of the even symmetry in both the frequency response and the weighting coefficient sequence. This method of obtaining the coefficients simplifies the coding considerably.

Following the design procedure outlined above, the cutoff frequency $f_c$was varied from 100 Hz to 12 Hz; many test data were passed through the filter. The pictures in Figures 4.4 and 4.5 obtained from the bit pad, were used as test figures. In order to evaluate the effect of filtering, the following strategy was used :

The cutoff frequency was progressively lowered until the smoothed pictures

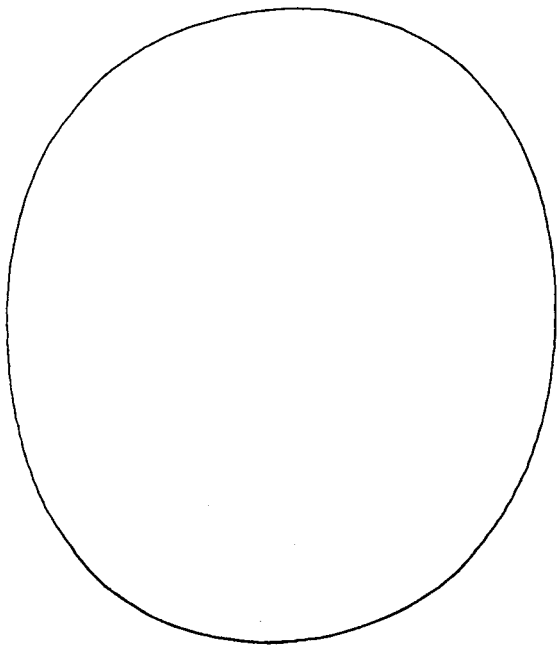


Fig.4.4
Test picture n0 1 generated
from the Bit Pad.

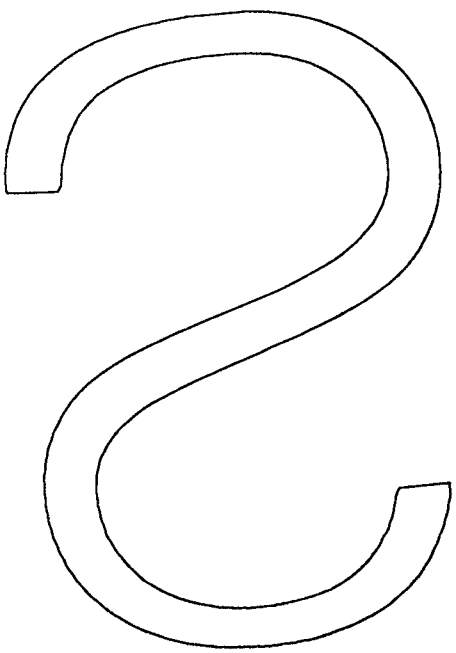


Fig.4.5
Test picture n0 2 generated
from the Bit Pad.

became unacceptable. The lowest cutoff frequency which produced acceptable pictures, was then retained as the ultimate frequency.

In so far as the filter characteristics are concerned, only the frequency response is shown, since the coefficients b_k have even symmetry,

i.e $b_k = b_{-k}$, the frequency response function is purely real, i.e the phase function is identically zero for all normalized frequencies and the phase shift between input and output sinusoids is zero and is independent of frequency.

Fig.4.6 depicts the frequency response of the filter 1, its cutoff frequency is 26 Hz. The number of terms of the filter is 32, of which 15 pairs of symmetric coefficients and the coefficient associated with the d.c conditions. The smoothed pictures generated by the filter 1, are shown in Fig 4.6.a, and Fig4.6.b, compared with the original, they are visually acceptable, so we can try to lower cutoff frequencies.

Filter 2 depicted in Fig.4.7, is characterized by a cutoff of 20 Hz. The number of the terms of the filter is 51, of which 25 pairs of symmetric coefficients and the coefficient associated with d.c conditions. The smoothed pictures generated by the filter 2 are shown in Fig.4.7.a and Fig.4.7.b, compared with the original, Fig.4.7.a is still very much the same as its original counterpart; but Fig.4.7.b starts to deteriorate at the regions of high frequencies, i.e corner regions.

Further lowering of the cutoff frequency to 14 Hz leads to further deterioration of the corner regions, in some applications such as typesetting, this is unacceptable, however in our application legibility is what matters. The deterioration is acceptable as long as the hand generated material

conveys the intended information. The frequency response of the filter is shown in Fig.4.8; the number of terms required was 44; visually Fig.4.8.a remains the same as its original, where as Fig.4.8.b is rather a poor version of the original.

The last cutoff frequency of 12.5 Hz leads to Filter 4, the number of terms is 41, of which 20 pairs of symmetric coefficients, and one term for d.c conditions. Clearly the result shown in Fig.4.9.b is unacceptable; fine details in the original drawing would have been smoothed out. However Fig.4.9.a is still acceptable. We have seen this pattern throughout; the reason for it is simple, that particular shape is largely made of very low frequency signals; if we inspect Fig.4.5 and its smoothed versions, i.e Fig.4.6.a, Fig.4.7.a, Fig.4.8.a and Fig.4.9.a, we undoubtedly see that the curved regions are still acceptable. This happens because they are regions of low frequencies, but the regions associated with the abrupt changes of directions are progressively distorted, and the distortion becomes unacceptable for cutoff frequencies below 14 Hz.

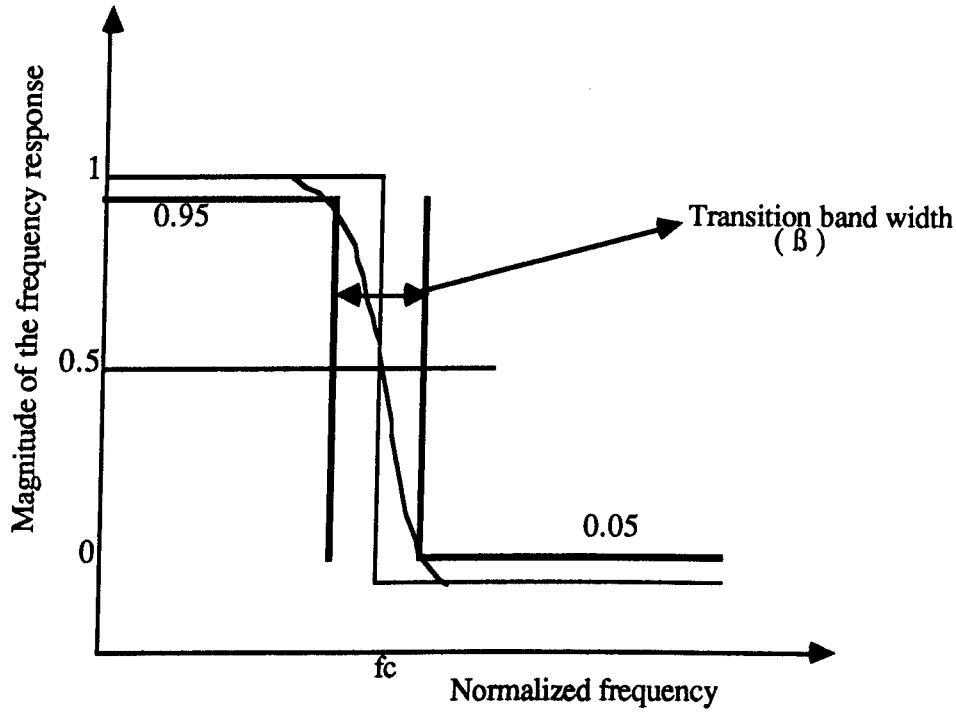


Fig.4.3 Ideal lowpass filter characteristic.

Given a cutoff frequency f_c, we try to model the ideal low pass filter in Fig.4.3 . Various approximations are shown in the following graphs obtained by suitably varying the cutoff frequency f_c.

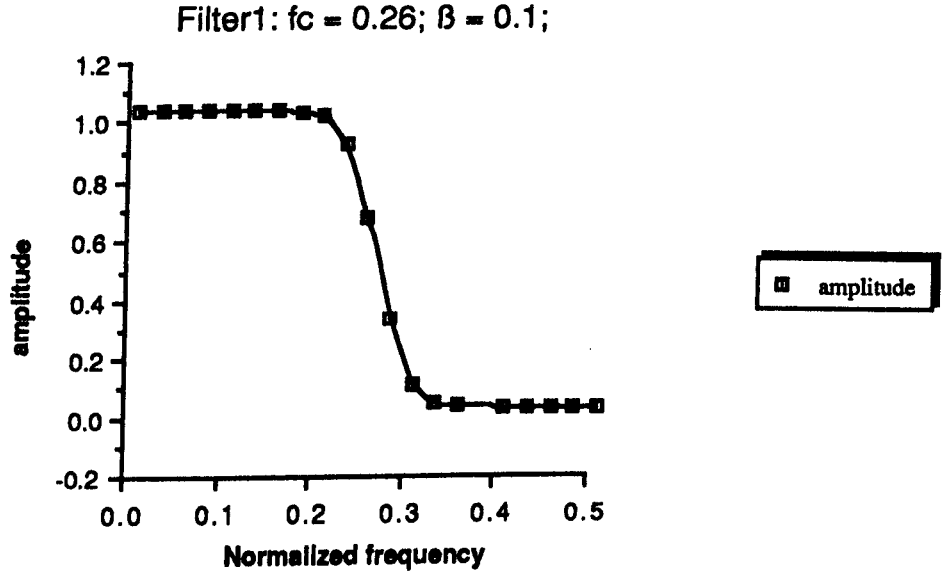


Fig.4.6 Filter 1 Frequency response

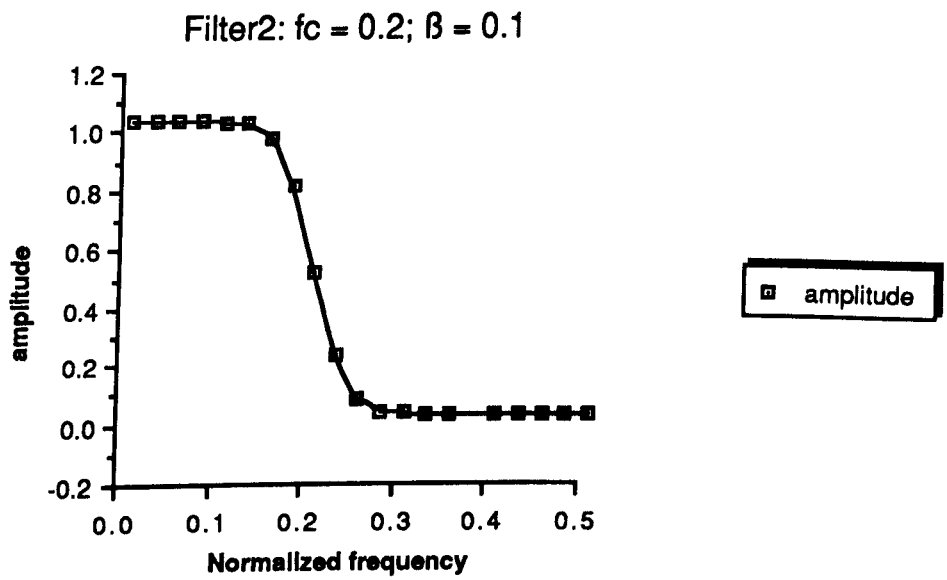


Fig.4.7. Filter 2 frequency response

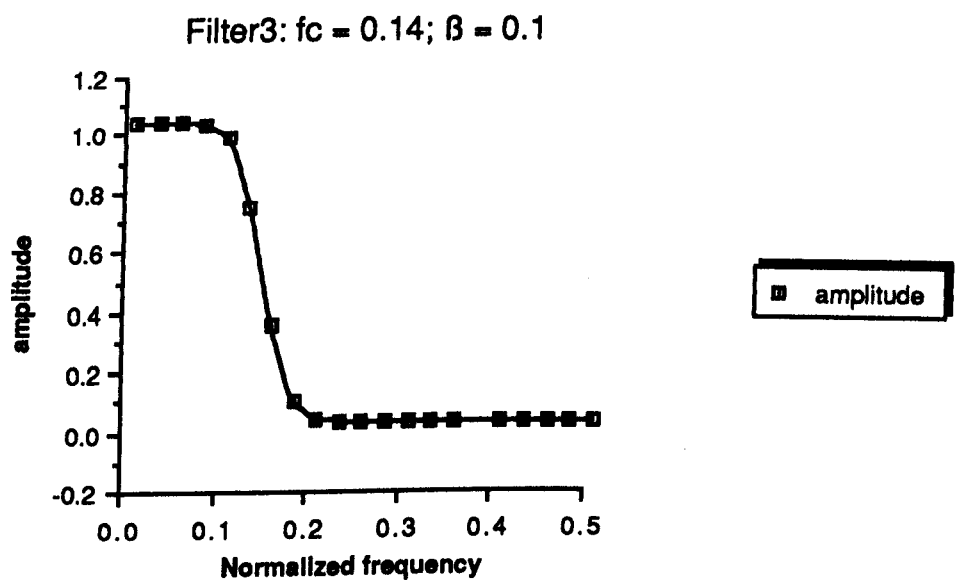


Fig.4.8 Filter 3 frequency response

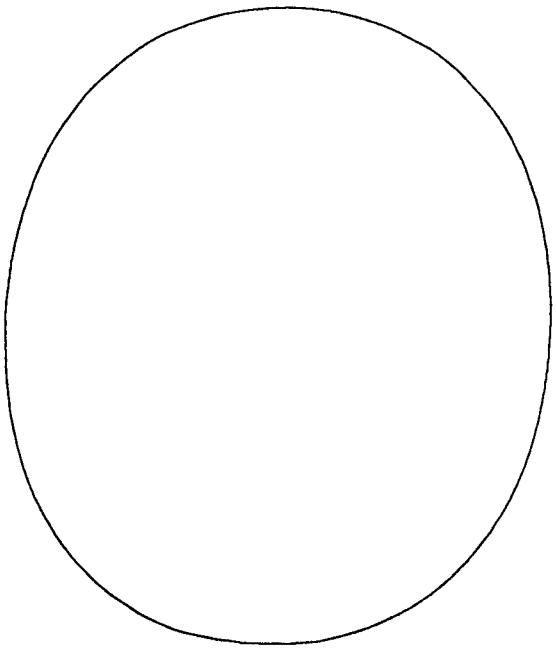


Fig.4.6.a
Lowpass filtered version of
test picture no 1
Filter characteristics:
Cutoff frequency = 26 hz
Number of filter coefficients = 32

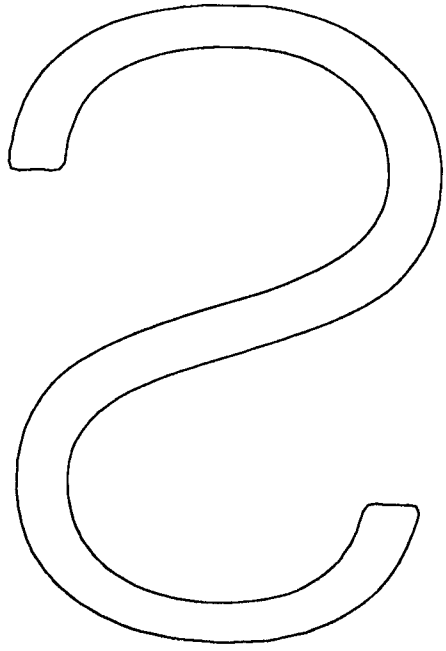


Fig.4.6.b
Lowpass filtered version of
test picture no 2
Filter characteristics :
Cutoff frequency = 26 Hz
Number of filter coefficients = 32

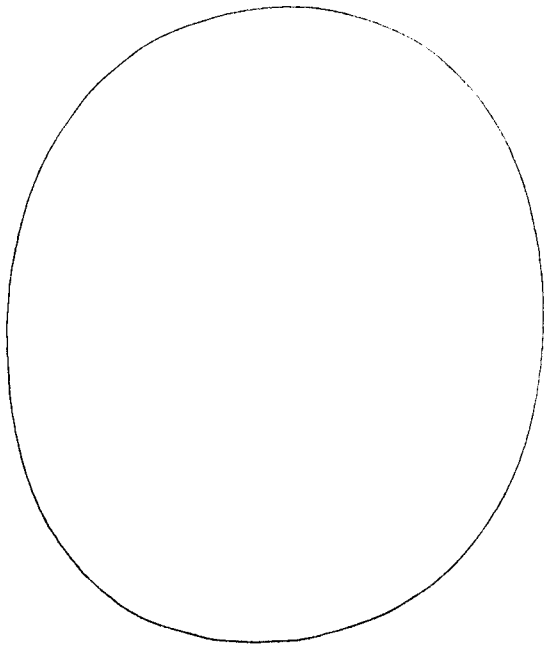


Fig. 4.7.a
Lowpass filtered version of
test picture no 1
Filter characteristics:
Cutoff frequency = 20 hz
Number of filter coefficients = 51

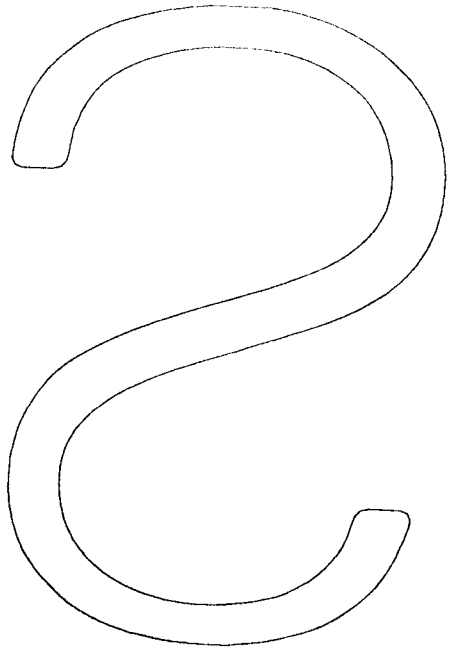


Fig.4.7.b
Lowpass filtered version of
test picture no 2
Filter characteristics :
Cutoff frequency = 20 Hz
Number of filter coefficients = 51

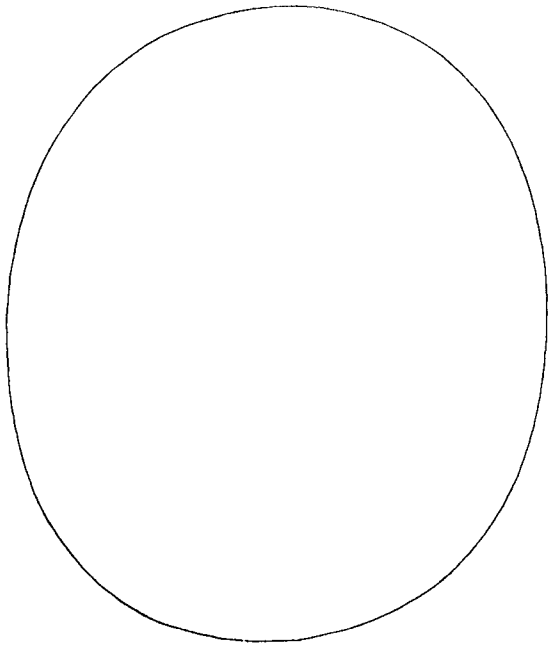


Fig.4.8.a
Lowpass filtered version of
test picture no 1
Filter characteristics:
Cutoff frequency = 14 hz
Number of filter coefficients = 44

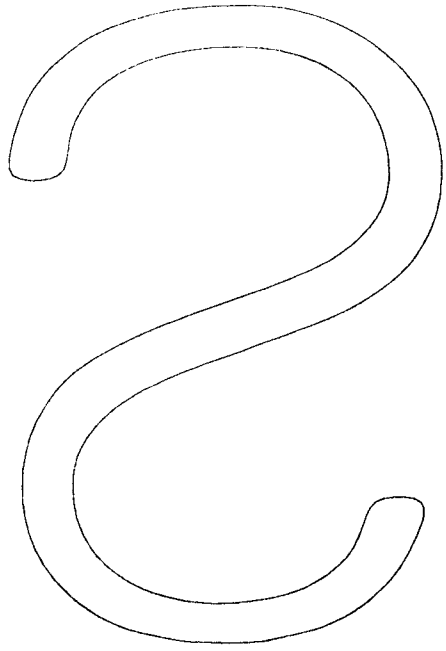


Fig.4.8.b
Lowpass filtered version of
test picture no 2
Filter characteristics:
Cutoff frequency = 14 hz
Number of filter coefficients = 44

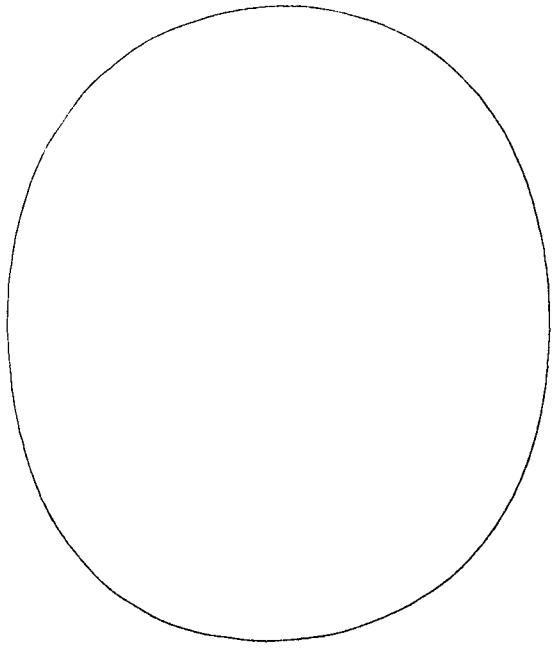


Fig.4.9.a
Lowpass filtered version of
test picture no 1
Filter characteristics:
Cutoff frequency = 12.5 hz
Number of filter coefficients = 41

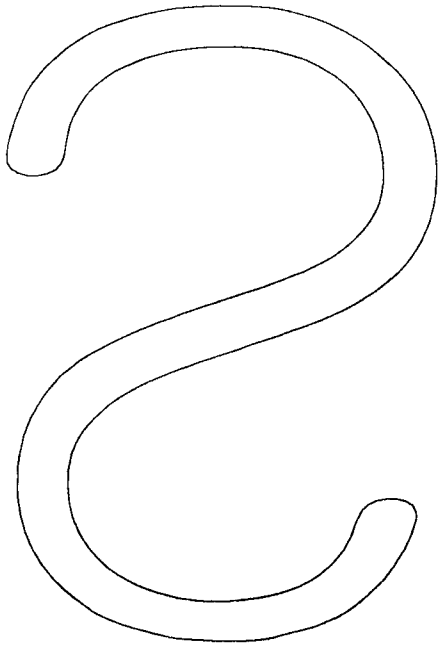


Fig.4.9.b
Lowpass filtered version of
test picture no 2
Filter characteristics:
Cutoff frequency = 12.5 hz
Number of filter coefficients = 41

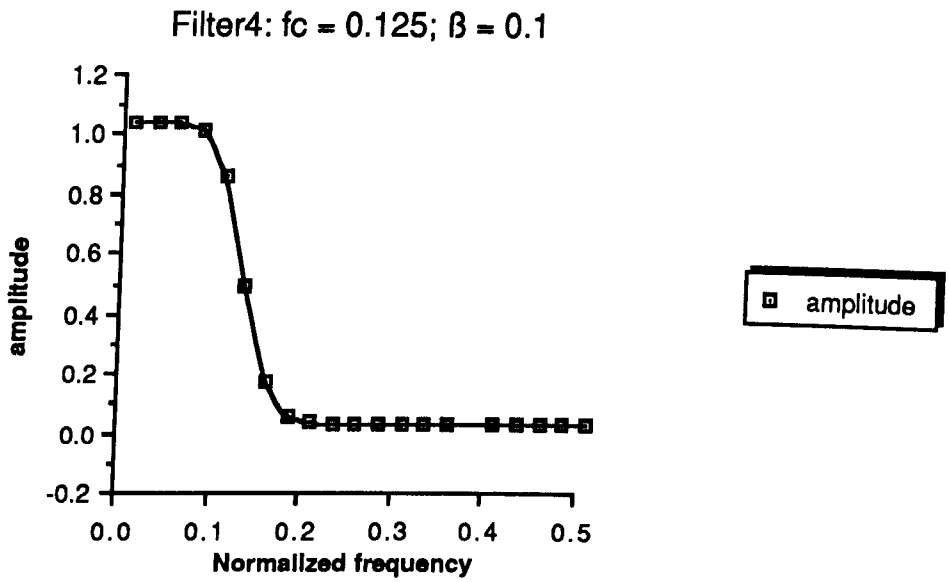


Fig.4.9 Filter 4 frequency response

Experiments conducted using digital filters on data obtained from various samples of hand generated material, have led us to discover that 14 Hz is a suitable effective cutoff frequency. Having determined the effective cutoff frequency, we can verify by decimating (i.e subsampling) the signal, this is is dealt with next.

4.3 Decimation

Decimation is the simplest technique available for coding hand generated material. It consists of taking the Dth sample point and capturing it as a data point (node). Like so many data tablets, ours operates at a fixed sampling rate (i.e 200 Hz). This rate is usually either under hardware or software control. Decimating (i.e subsampling) then simply reduces the effective sampling rate, relying on an interpolation algorithm at the receiving terminal to reconstruct the original picture (SCHAFER73).

The output of the digital low pass filter was fed to the decimator as depicted in Fig.4.2. Experiments were conducted on the handwriting and drawing

of twenty one subjects. The synthesis of handwriting and drawing from decimated signals was investigated for decimation factor ranging from 1 to 28. Of course the decimation factor of 1 corresponds to the original smoothed data. The quality of most of the reconstructed pictures, was poor when D was greater than 8.

A suitable maximum decimation factor for handwriting and drawing was found to be 7. This figure was determined by conducting experiments on the drawing and handwriting of thirteen different subjects.

Typical results are shown in the following figures :

Fig.4.10 is the original graphic material reproduced using 200 samples per second. The smoothed and decimated versions of that material are shown in Fig.4.11, Fig.4.12, Fig.4.13.

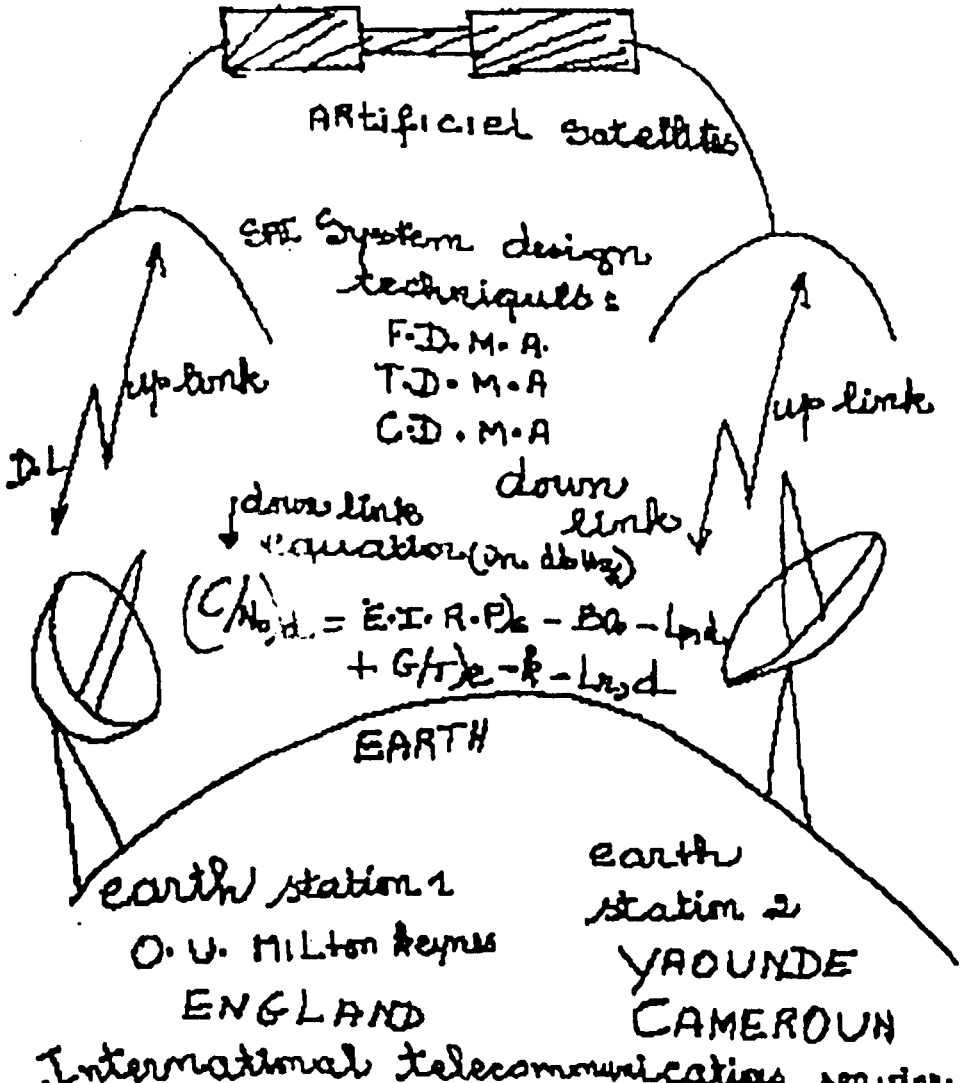
The total number of points making the original picture (Fig.4.10) was 89407. Fig.4.11 only has 6877 points. This represents a 13 to 1 decimation. Most of handwriting has become illegible, whereas the drawing conveys the intended information adequately (i.e message).

Fig.4.12 has 9934 points, which represents a 9 to 1 decimation. The drawing and handwriting are much more acceptable. In fact, Fig.4.12 is better than the original because the wobbles are smoothed out.

Fig.4.13 consists of 12772 points, this represents a 7 to 1 decimation, the drawing and handwriting are clear enougth to be acceptable by most people. Of course, the material we have just seen, comes from one subject, and different people have different styles of handwriting. Having run the experiments on thirteen tutorials from thirteen different subjects, we have found that for a 7 to 1 decimation , the results were very good, and, from the perceptual point of view, the pictures presented to the observer were almost unchanged, compared to the original counterparts.

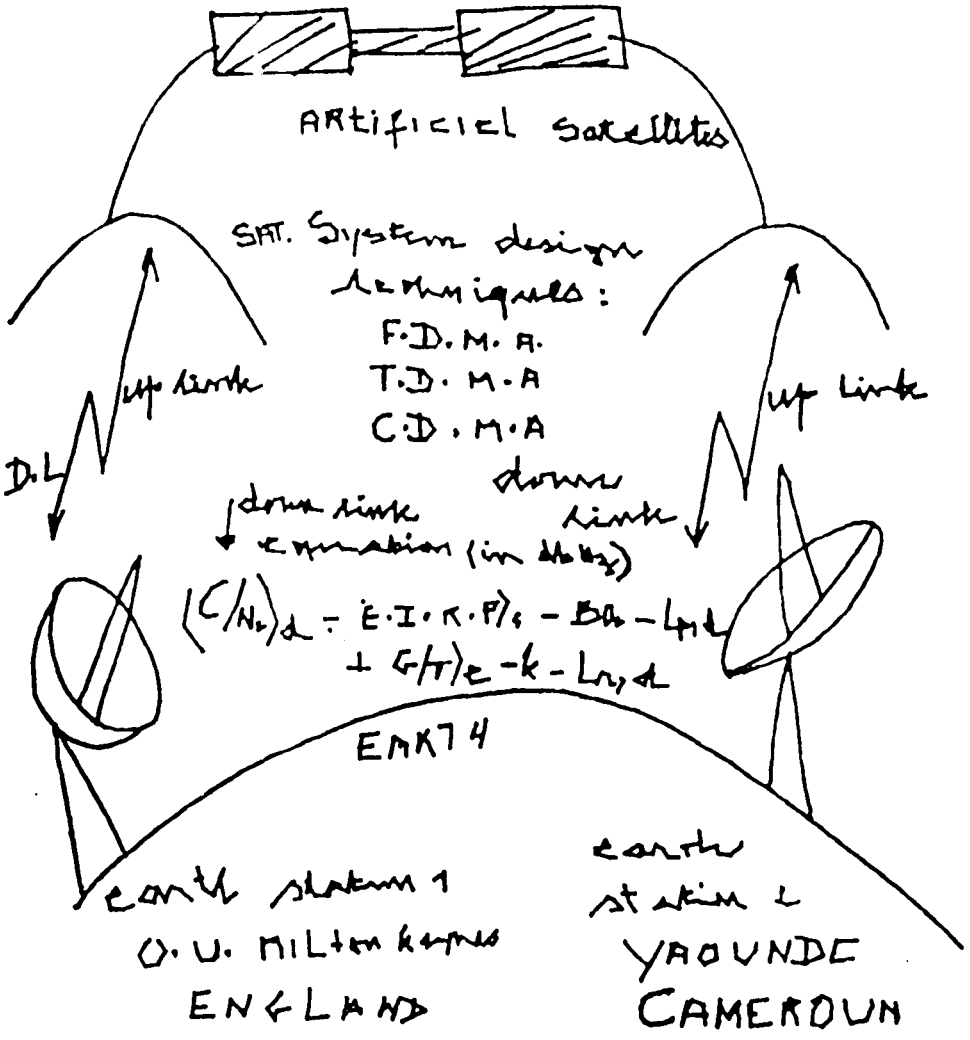
Fig. 4.10

Original graphical material reproduced using 200 samples per second



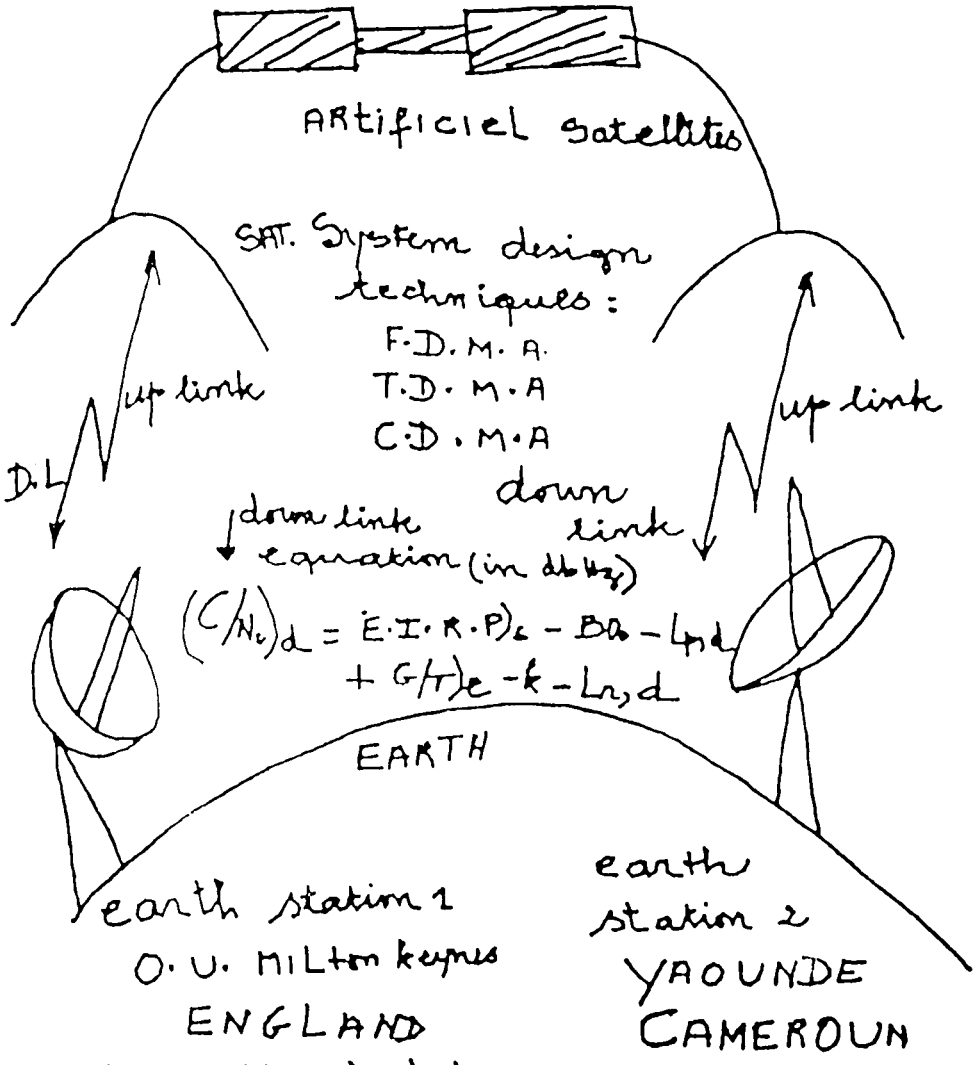
International telecommunications services have experienced tremendous growth since satellites took off in 1965.

Fig.4.11
Decimated version of graphical material .
Decimation factor = 13



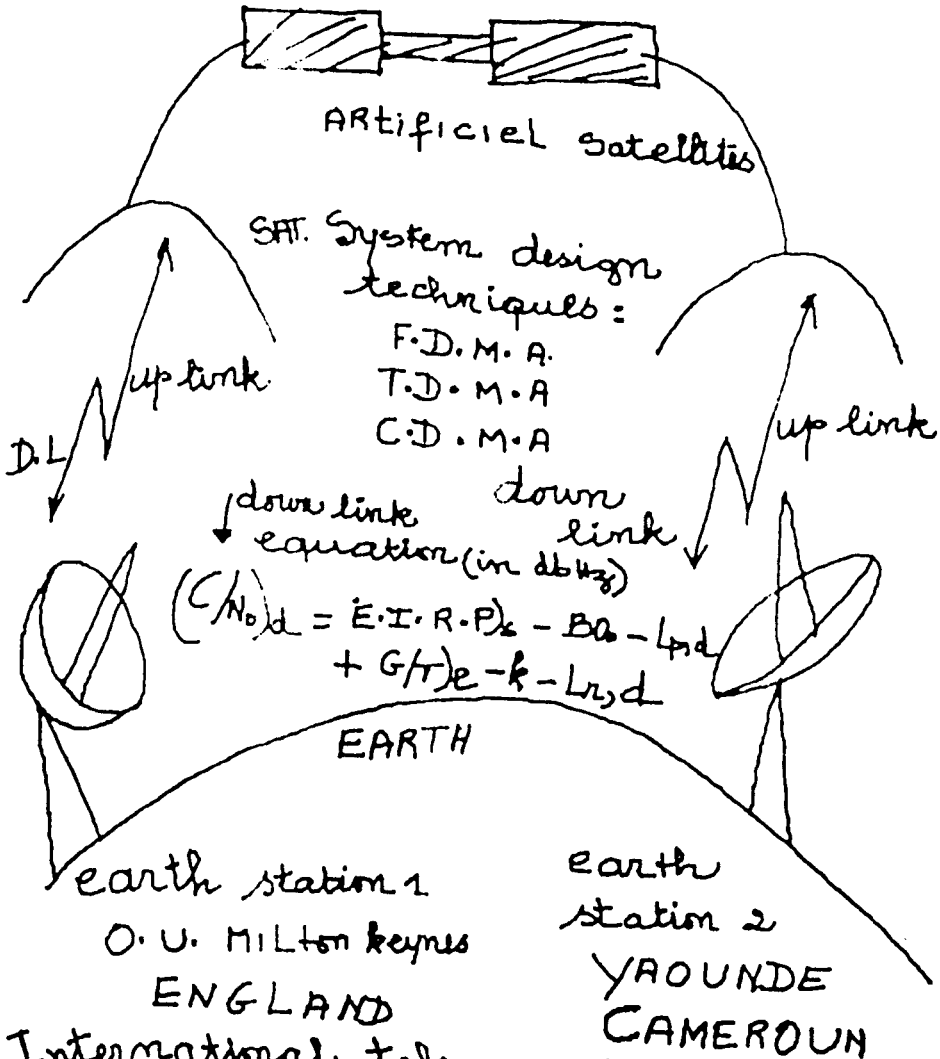
International telecommunications services have experienced tremendous growth since satellites took off in 1965.

Fig.4.12
Decimated version of graphical material.
Decimation factor = 9



International telecommunications services have experienced tremendous growth since satellites took off in 1965.

Fig.4.13
Decimated version of graphical material.
Decimation factor = 7



International telecommunications services have experienced tremendous growth since satellites took off in 1965.

The reconstruction process was done by joining subsampled points by straight line segments, i.e straight line interpolation.

The subsampling method has many advantages. It is robust and simple. In our application, the data channel has a fixed bandwidth, thus the data transmission rate can be matched to the channel capacity. With a subsampling system, the user has some control over the accuracy of the coding. When finer detail is required, the picture is simply drawn more slowly. This dependence on the user physiology also works to good advantages near corners. The natural motion is for the pen to slow down (at corner regions), so that the points become more densely packed. The subsampling method is also fairly scale independent. For example, the time required to draw a symbol, and hence the number of points used to code it, is only slightly dependent on its size.

What disadvantages does this method have ? First and foremost, it does not yield optimally compressed pictures. When higher compression is important, such as with database storage, one could couple the subsampling technique and a shape dependent technique to produce better results; this will be looked into in the next chapters.

The subsampling method is also very much hardware or implementation dependent. The sampling rate of the graphics device determines the final data rate. As well, the degree of user control exercised by the speed of writing can be a disadvantage, especially with inexperienced or hesitant users.

On the whole, however, subsampling produces good results. A visual study of various pictures shows that the results degrade gracefully as the sampling rate decreases.

4.4 Concluding comments on previous sections

Our signal was previously digitized with a high sampling rate of 200 Hz. It was then digitally lowpass filtered to remove frequency components lying above the highest frequency of interest 14 Hz. The output of the digital lowpass filter was resampled by preserving only every seventh output number and discarding the rest. This decimation factor was found after many trial on various hand generated pictures. It is most satisfying to realize that the decimation factor of 7, producing, in principle, a cutoff frequency of 14.28 Hz, corresponds almost to the cutoff frequency of our lowpass FIR filter, 14 Hz, which was found by carefully controlling the parameters of the lowpass filter, and evaluating the results visually. Having determined the decimator factor, all our tutorials were subsampled using the technique described above. The resulting data were statistically analyzed in the spirit of chapter 3, as described in the next section.

4.5 Statistics of the decimated hand produced digital signals

As in chapter 3, we are going to describe the estimates of the frequency of occurrence of difference digital signals (i.e Δx or/and Δy), n-gram probability distributions and different measures of entropy. The purpose of the entropy of the signal was sufficiently made clear in chapter 3, and does not need to be repeated here.

Compared with chapter 3, the only difference is that, the data representing each subject's handwriting has been significantly reduced by the process of 7 to 1 decimation. The frequency distributions of $\Delta x/\Delta y$ are shown in Table.4.1.a. Using them we can estimate the first order entropy, in bits per quantized difference signal in horizontal and vertical direction, and adding them up leads to the estimate of the first order entropy of the difference signal. The same strategy is used to find the higher order estimates of the

signal. With reference to chapter 3, we recall the definition of the n-gram entropy for the quantized difference signal.

$$H_n = -\sum_{j1\,,\,j2\,,\,j3\cdots\cdots jn\,,\,j} p(\Delta_{j1}, \Delta_{j2} \ldots\ldots \Delta_{jn}\,,\,\Delta_j)\ \log_2 p(\Delta_j \mid \Delta_{j1}, \Delta_{j2}\ldots\ldots\Delta_{jn}) \qquad (4.6)$$

where $\Delta_{j1}, \Delta_{j2} \ldots\ldots \Delta_{jn}$ denote a sequence of n pen movements Δs

The estimates of the entropy of the decimated signal are given in Table 4.2 and Table 4.3. To obtain the estimates of the probabilities needed in equation (4.6), the whole database, made of all the available tutorials, was fed to the filter 3 discussed above; for every 7 filtered outputs from the filter, one was retained; in this way we created a set of data which were appropriately differenced. The frequency distributions of the differences were then measured as explained in chapter 3. The relative frequencies were then used as estimates of the required probabilities.

Analysis of the results:

The first order distributions curves for Δx , Δy are shown respectively in Fig.4.14 and Fig.4.15.

In so far as trends are concerned, they are very similar to the curves found, when statistical analysis was carried out on original data (see chapter 3); however the number of classes of Δ has increased almost four fold. As in chapter 3, the distributions follow an exponential trend and are highly peaked at $\Delta = 0$. The higher the absolute value of Δ, the lower the corresponding relative frequency; in other words, increasing absolute values of classes of Δ become more and more rare.

Table 4.2 shows the theoretical and practical estimates in bits. Column 1 indicates the order of correlation. The maximum entropy estimate occurs for independent and equiprobable Δ's; this means that the Δ's have flat probability distributions. As Δx or Δy varies from -40 to +44, 85 classes of Δ result from either x direction or y direction. So the zeroth approximation of entropy consists of a source of 85 elements, all equiprobable, providing

12.80 bits per Δ (i.e $\log_2 85 \approx 6.4$ bits per Δx or Δy).

Using the correct Δ probabilities, the first approximation of entropy is 5.35 bits per Δ.

Using the conditional probabilities over pairs of adjacent (Δx or Δy), the second entropy approximation is 4.34 bits.

A third approximation, in which conditional probabilities over three Δ is 3.91 bits.

The results of Table 4.2, suggest that, from third entropy approximation, the effect of correlation (on entropy estimate) becomes smaller and smaller. In other words, the higher the order of correlation, the smaller the difference between successive estimates of entropy. This pattern agrees with observations made in the previous chapter.

From chapter 3, we know that the redundancy is given by

Redundancy = (maximum entropy estimate - lowest entropy estimate)

Expressing this in terms of the order of correlation, n, we have

$$R_n = (H_0 - H_n) \text{ in bits}$$

or

$$\text{Relative } R_n = ((H_0 - H_n)*100)/H_n \text{ in percentage}$$

where H_0 is the maximum entropy estimate and H_n the lowest entropy estimate associated with nth order correlation between n successive Δ's or

n+1 successive pen positions.

Applying these formulae, we can construct the following table for various orders of correlation:

Order of correlation	Redundancy in bits	Relative redundancy in percentage
0	0	0 %
1	6.84	52 %
2	7.68	58 %
3	8.01	60 %
4	8.27	62 %
5	8.48	64 %
6	8.66	65 %
7	8.80	66 %
8	8.92	67 %

The graphical presentation of the redundancy is shown in Fig.4.18.

Fig.4.18 shows the increase in redundancy as successively more past pen positions are taken into consideration. The ordinate at n = 1 is the previous pen position redundancy, while n = 2 gives the redundancy due to the two previous pen positions. The ordinate at n = 8 is the redundancy in the 8 pen positions closest to the present pen position. It is seen that the redundancy levels off very rapidly so that the the previous-pen position redundancy is a very significant portion of the total redundancy.

Given a 7 to 1 decimation, the results of Fig.4.18 suggest that the redundancy be expected to lie between 6 and 9 bits.

Table 4.3 shows the results of interests. It is clear that data reduction by the process of decimation leads to encouraging entropy rate estimates. We do know that the target bit rate is 200 bits per second, but if we can do better, i.e

having a bit rate significantly lower than 200 bits per second, this is well worth achieving because the final coding is likely to provide for error detection and correction, this means that the final channel coded bit rate might be, significantly higher. So, to be in the safe side, having an entropy rate estimate as low as possible is welcome. Of course, we mean the lowest entropy estimate which would produce a faithful reproduction of the picture
(i.e the intended written material should be conveyed without any ambiguity).

The theoretical limits of entropy rate estimates are shown in
Table 4.3 (columns 2 and 3). To obtain the practical limits (column 5 of Table 4.3), we can use the statistics of the pen runs obtained from the decimated signal, which are shown in Table 4.1.b, Figures 4.16 and 4.17.
Fig.4.16 depicts the frequency distribution of the pen run lengths, when the signal is subject to a 7 to 1 decimation. Fig.4.17 depicts the corresponding cumulative probability distribution. The main results conveyed by Table 4.1.b, Figures 4.16 and 4.17 are:

1. The smallest pen run length is 1 sample.
2. The largest pen run length is 396 samples.
3. The average pen run length is 11 samples;
4. 70 % of lengths are less than 11 samples (i.e pen run length average).
5. 90 % of lengths were less than 22 samples.
6. 95 % of lengths were less than 33 samples.

When we compare these results with the ones described in chapter 2, they agree with our expectations. For example the average length before the decimation was 80 samples; after decimation the average length became 11 samples.
As expected, this corresponds to 7 to 1 decimation.

Assuming a writing surface of 396 by 522, we use 18 bits to represent each point considered isolated from the previous and subsequent points. As there were 14557 traces, 14557 points carrying 18 bits are all starting points of the traces. Using the same strategy as explained in chapter 3, the practical limits of the estimates of the entropy obtained from the decimated signal were:

a). 379 bits per second when zero order entropy rate of the difference signal was considered. Here the distributions of Δx or Δy were assumed flat; i.e all the possible classes of Δx or Δy were assumed equiprobable.

b). 183 bits per second when only first order entropy rate of the difference signal, was considered. This first order approximation of entropy rate consisted of putting in the correct probabilities of Δx and Δy.

c). 159 bits per second when first and second order entropy rate estimates of difference signal, were considered.

d). 150 bits per second when first, second and third order entropy rate estimates of the difference signal, were considered.

e). 142 bits per second when first, second, third and fourth entropy rate estimates of the difference signal, were considered.

g). 136 bits per second when up to fifth order entropy rate estimates of the difference signal, were assumed.

h). 131 bits per second when up to sixth order entropy rate estimates of the difference signal, were assumed.

i). 127 bits per second when up to seventh order entropy rate estimates of the difference signal, were assumed.

j). 124 bits per second when up to eighth order entropy rate estimates of the difference signal, were considered.

If we look at the combined signal entropy rate (practical limits), we see that the first order entropy rate is 183 bits per second, this is 8.5 % below 200 bits per second, the second order entropy rate of 159 bits per second is 20.5 % below 200 bits per second, the third order entropy rate of 150 bits per second is 25 % below 200 bits per second.

The fourth to eighth order entropy rates are respectively 29 %, 32 %, 34.5 %, 36.5 %, 38 % below 200 bits per second.

Practically, under a 7 to 1 decimation process, the minimum achievable bit rate is 124 bits per second, if one compares it with the original 11000 bits per second (see chapter 3), we are talking about a substantial redundancy; almost every second, 10876 bits do not need to be sent, to recover a perceivably acceptable picture. The entropy rate of 124 bits per second has been possible because the filtering has reduced unwanted detail and noise.

Δx/Δy	p(Δx)	p(Δy)
-20	0.00005147	0.0000257
-19	0.00005662	0.0000514
-18	0.00006691	0.0000669
-17	0.00009265	0.0000761
-16	0.00012456	0.0001286
-15	0.00014567	0.0001595
-14	0.00018016	0.0001956
-13	0.00018531	0.0003303
-12	0.00018531	0.0005456
-11	0.00030371	0.0007721
-10	0.00038092	0.0012148
-9	0.00052505	0.0015288
-8	0.00070522	0.0022752
-7	0.00093686	0.0041335
-6	0.00138986	0.0064499
-5	0.00267676	0.0115821
-4	0.00580653	0.0223304
-3	0.01414055	0.0438063
-2	0.03546205	0.0859346
-1	0.09938537	0.1656199
0	0.44076617	0.4457336
1	0.18792982	0.1139325
2	0.09497899	0.0459580
3	0.04648313	0.0214707
4	0.02395708	0.0110159
5	0.01335811	0.0065117
6	0.00797368	0.0034334
7	0.00516822	0.0018840
8	0.00368055	0.0009058
9	0.00295988	0.0006177
10	0.00231643	0.0004941
11	0.00169357	0.0002985
12	0.00121484	0.0001595
13	0.00093172	0.0000823
14	0.00067434	0.0000772
15	0.00062286	0.0000617
16	0.00036548	0.0000617
17	0.00033453	0.0000154
18	0.00031400	0.0000051
19	0.00025223	0.0000051
20	0.00016987	0.0000051

Table 4.1.a first probability distributions
for Δx and Δy obtained from decimated signal

Fig.4.14

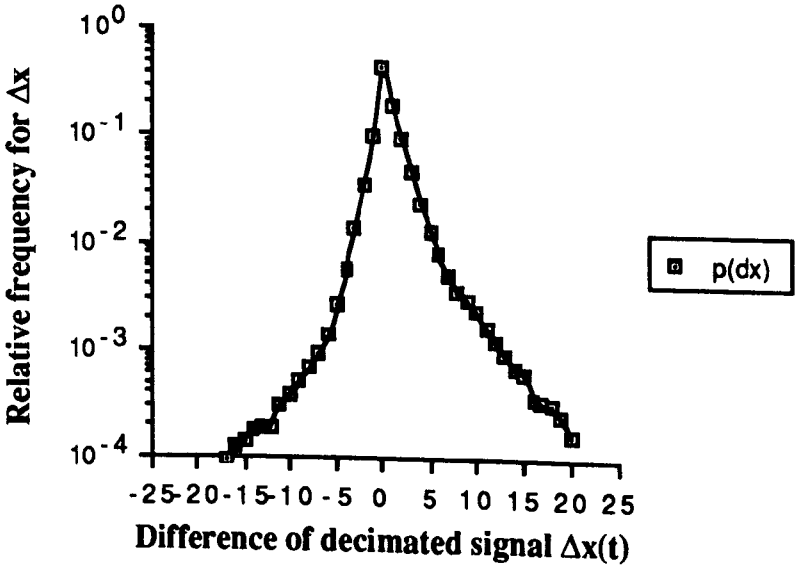
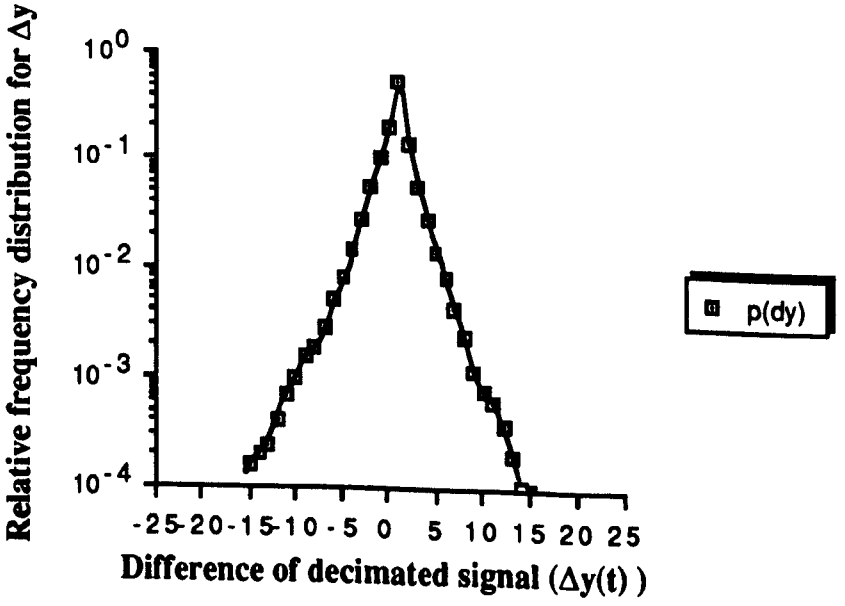


Fig.4.15



length	frequency	length	frequency	length	frequency	length	frequency	length	frequency
1	34	2	636	3	1407	4	1508	5	1238
6	1125	7	1063	8	910	9	861	10	621
11	523	12	467	13	460	14	390	15	368
16	311	17	235	18	218	19	184	20	185
21	174	22	116	23	97	24	94	25	110
26	72	27	73	28	68	29	56	30	50
31	40	32	38	33	32	34	46	35	36
36	35	37	31	38	31	39	36	40	26
41	20	42	19	43	11	44	25	45	20
46	27	47	16	48	20	49	24	50	16
51	17	52	7	53	15	54	15	55	11
56	9	57	12	58	11	59	15	60	10
61	6	62	9	63	7	64	10	65	7
66	2	67	10	68	10	69	4	70	5
71	6	72	6	73	2	74	7	75	5
76	5	77	3	78	4	79	3	80	3
81	3	82	4	83	3	84	8	85	3
86	5	87	2	88	4	89	7	90	1
92	2	93	1	94	1	95	2	96	2
97	2	98	3	99	1	100	1	101	1
102	2	103	5	104	3	105	2	109	2
111	1	112	3	116	1	117	1	118	2
120	2	122	2	123	1	125	2	127	1
129	1	131	1	133	2	139	2	140	1
141	1	145	1	147	1	148	1	151	1
157	1	159	1	160	1	163	1	165	1
166	1	175	1	177	1	180	1	182	1
184	1	190	1	196	1	202	1	213	1
221	1	224	1	225	1	369	1	396	1

Table 4.1.b statistics of the pen runs after a 7 to 1 decimation

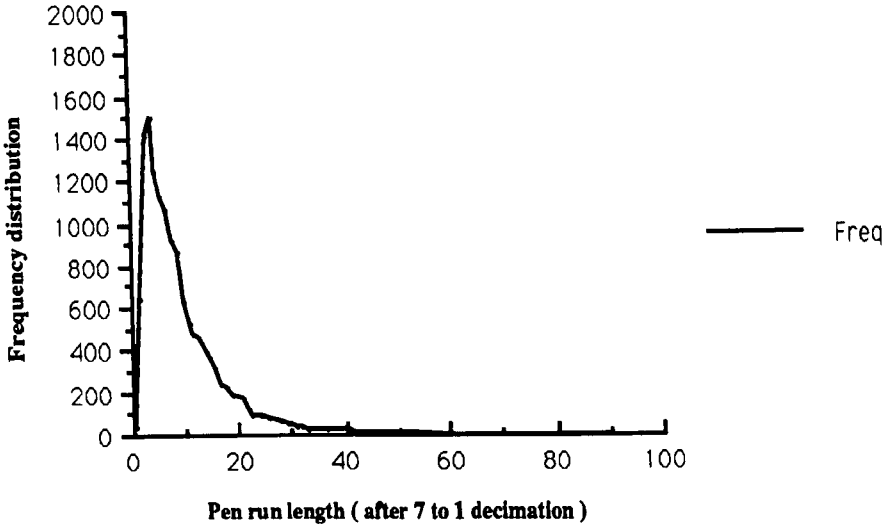


Fig.4.16 Distribution of pen run length, after 7 to 1 decimation.

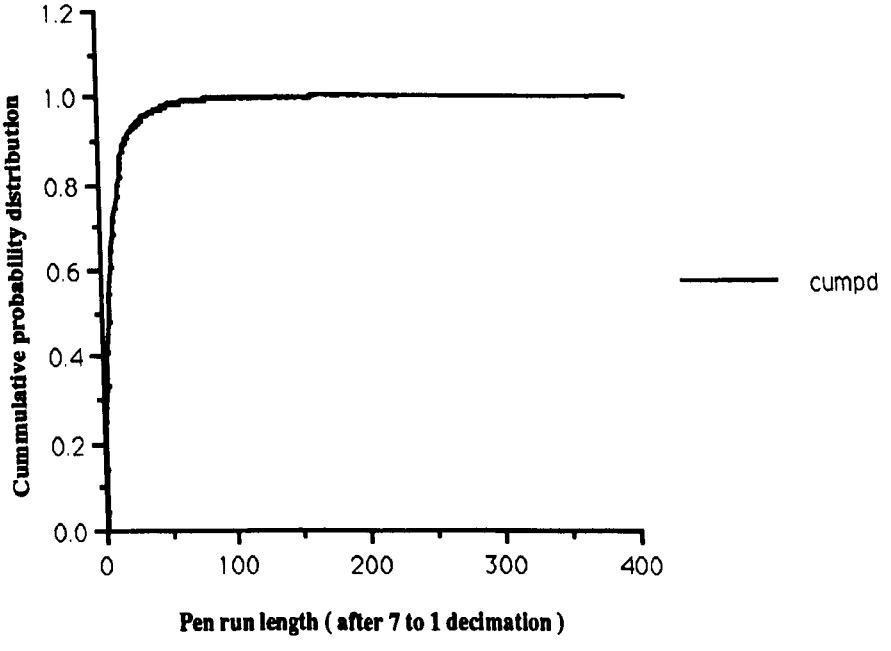


Fig.4.17 Cumulative probability distribution (after 7 to 1 decimation)

Entropy estimates of the decimated signal (in bits)				
Order of correlation	Δx signal	Δy signal	Combined signal (Theoretical)	Combined signal (practical limit)
0	6.40	6.40	12.80	13.25
1	2.68	2.67	5.35	6.41
2	2.16	2.18	4.34	5.57
3	1.97	1.94	3.91	5.24
4	1.80	1.72	3.52	4.98
5	1.64	1.54	3.18	4.77
6	1.50	1.37	2.87	4.59
7	1.37	1.22	2.59	4.45
8	1.25	1.08	2.33	4.33

Table 4.2 Entropy estimates of decimated signal (7 to 1 decimation)

Entropy rate estimates of the decimated signal (in bits per second)				
Order of correlation	Δx signal	Δy signal	Combined signal Theoretical	pratical limit
0	200	200	400	410
1	76.57	76.48	153.05	183
2	61.82	62.45	124.27	159
3	56.28	55.45	111.73	150
4	51.43	49.15	100.58	142
5	46.86	44	90.86	136
6	42.86	39.15	82	131
7	39.15	34.86	74	127
8	35.71	30.86	66.57	124

Table 4.3 Entropy rate estimates of decimated signal (7 to 1 decimation)

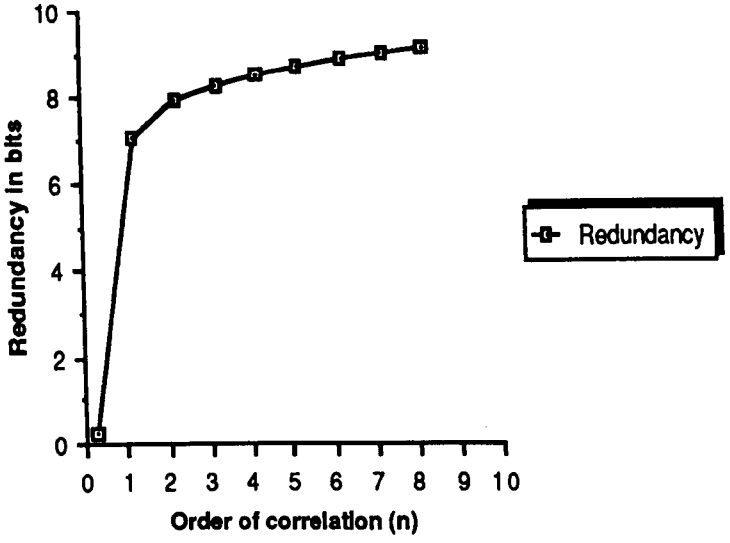


Fig.4.18 graphical presentation of redundancy versus correlation order.

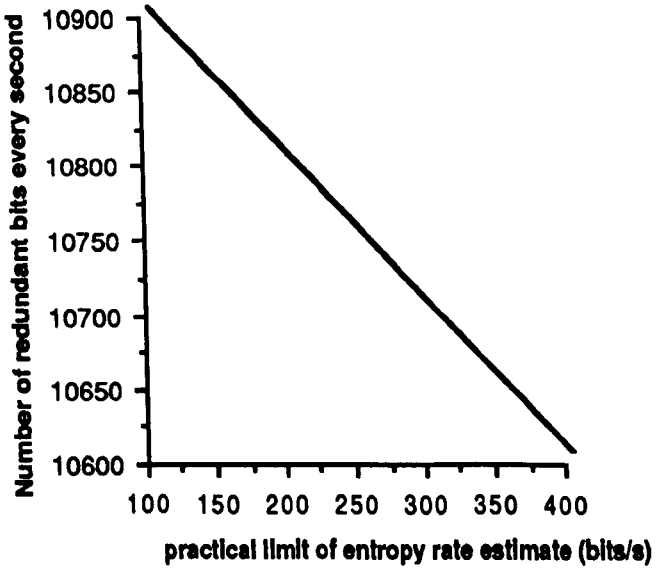


Fig.4.19 Graphical presentation of redundant bits/s versus entropy rate.

If a coding scheme is devised to utilize the previous-pen position redundancy only, the results of Fig.4.8, suggest that the incorporation of higher order redundancies would yield rapidly diminishing returns, probably not enough to justify the complexity of the coding scheme.

The graphical presentation of the redundancy per second is shown in Fig.4.19, which suggests a linear relationship between the entropy rate estimates and the number of redundant bits per second.
A least error squarred fitting produces Rbs = 11000 - h, where h is the practical entropy rate estimate.

A survey of the overall graphic material has shown that there are 1283876 points before the decimation, and 210953 points after the decimation; this corresponds to data point reduction rate of about 84 per cent. If we were storing those points, on the basis of 20 bits per point, about 26 megabits of storage space would be needed. This storage requirement would drop to 4.219060 megabits because of the decimation.
As far as the n-gram distribution of deltas is concerned, it was found that there were 194264 unigram of deltas (Δx or Δy), i.e deltas are considered independently; for the bigram (i.e second order correlation), there were 178300 pairs of deltas (Δx or Δy) and 163906 triagrams, i.e third order correlation, of deltas. It was found that the higher the correlation order n, the lower the number of n-tuples of successive (Δx and/or Δy).

4.5.1 So far, how good are the entropy rates measurements ?

The graphical comparison of entropy rate estimates measurements, is shown in Fig.4.20, where there are three curves :

1. The flat curve is the target bit rate (i.e 200 bits/s) curve, and is our reference curve. In our work, the forbidden zone is above the reference curve, whereas the allowed zone (i.e accepted zone) is below the reference curve.

2. The top curve is the graphical presentation of the entropy rate estimates measured from the original data (see Table 3.6 of chapter 3). The lowest point of the curve corresponding to 8th order correlation, is above the target bit rate; so as observed in the previous chapter, this is unacceptable.

3. The bottom curve is the entropy rate curve associated with the decimated signal. We can see that, from first order correlation, the entropy curve progressively deepens below the target bit rate curve.

The entropy rates measured from the decimated signal are encouraging, the lowest entropy rate estimate is 124 bits/s (order of correlation, n = 8). Previously, we have observed that from second order correlation, redundancies converge rapidly (see Fig.4.18), and that the redundancy associated with first order correlation constitutes a very significant part of the total redundancy. This suggests that designing a coding scheme which exploits all nth order redundancies, is worthless, because the complexity of the coding system may not justify the small contributions of higher order redundancies to the total redundancy.

Thus, it is thought that increasing the first order redundancy is appealing because, it may mean a significant lower bit rate than 183 bits/s using only first order correlation. Furthermore, constructing a code using only first correlation (between successive pen positions) may be easier.
We think that, if a significant lower entropy rate is to be realized at first order correlation between successive pen positions, it must come from two aspects of the redundancy :

1. Statistical redundancy. This is what we have analyzed so far.
2. The non statistical redundancy. To reproduce a pen trace with more accuracy than necessary is to supply redundant information. This type of redundancy is virtually independent of the statistical characteristics of the signal. One must devise intelligent ways to "fool the eye", -to transmit degraded pen traces with the degradation inserted so that it will not be detected by the human observer. So the way to exploit the non statistical redundancy is by cleverly degrading the " information " in the pen traces in such a way that the human observer would not notice it appreciably. Here the word " information " is not employed in the entropic sense, because with reference to SHAN48, entropic information rate can only have magnitude, not quality; it cannot be "degraded".

In the later chapters, the non statistical ("a better word may be physiological") redundancy, coupled with the statistical redundancy will produce significantly lower "first order correlation " entropy rate. From the point of view of coding, this is an advantage, because it may imply a simpler coding scheme based only on first order statistics of the pen movements.

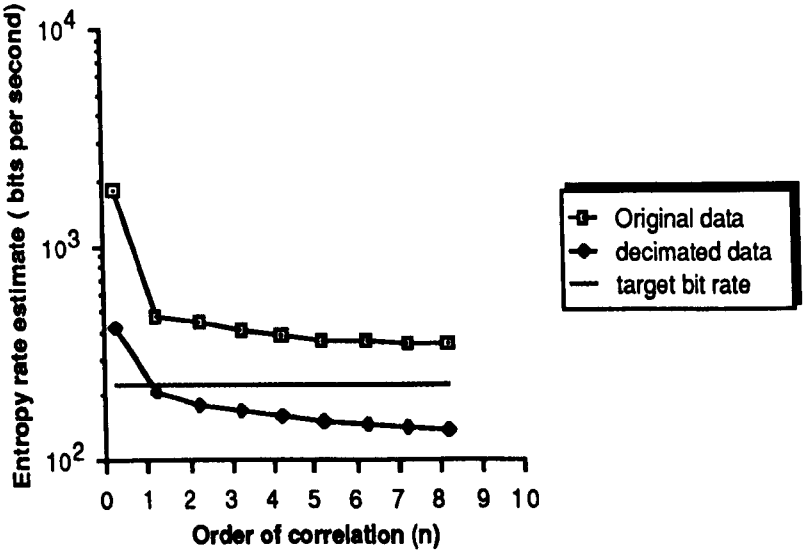


Fig.4.20 Graphical comparisons of entropy rate curves

4.6 Conclusions

Under real time conditions, it has been found that the handwriting signal can be sampled about at 29 Hz, and can be reconstructed faithfully. This implies that the useful bandwidth of the handwriting and drawing signal is about 14.5 Hz. An appropriate selection of the cutoff of a non recursive filter led to a suitable frequency of 14 Hz below which the some pictures were significantly distorted. This cutoff was checked by a 7 to 1 factor decimator, and the result was effectively the same.

The statistics of the decimated signal were measured. With respect to the entropy of the original quantized signal, without any decimation (see chapter 3), there has been a significant drop in the entropy rate of the signal. The figures speak for themselves. Without decimation an entropy rate estimate of 314 bits per second was found; with a 7 to 1 decimation, an estimate of 124 bits per second was found, this represents 60.5 % drop.

The analysis of the statistical redundancy provides the evidence that the higher order redundancies converge rapidly, and that the first order redundancy constitutes a very significant part of the total redundancy.

A variable subsampling of the decimated signal will bring down the entropy rate of the signal further, this is explored in the following chapters, where a 7 to 1 decimated signal is used as a baseline. In this chapter the data reduction technique used is essentially time related, in the following chapters, the criteria for reducing the data further is based upon the spatial data describing the trace of the writing pen.

5. REAL TIME ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT DIGITIZED FREEHAND PRODUCED DRAWING AND HANDWRITING

The previous chapter was concerned with the optimum temporal sampling of the hand generated data. Assuming that the Nyquist frequency was successfully determined, the data which represent the picture signal could be sampled at just twice that frequency, and the original picture could be represented inside the computer in terms of in terms of geometric primitives be they straight line segments, arcs or special curves such as splines (SCHAFFER73).

The last chapter indicated some of the limitations of the decimation technique. Simple time sampling may not capture the cusps and line endings which are characteristics of the handwriting. A second limitation is that straight lines are over-represented. A third limitation is that duplicate points will occur.

The alternative to deleting points is to select points. This is the first of the chapters in which we attempt to give suitable answers to the following question :

How do we reduce the number of points in a pen trajectory and yet retain the shape of the original trace, especially in handwriting, which may be mostly characterized by curved traces ?

Preservation of the information in the pen traces, " the significant features", is the primary consideration. Thus fidelity must be traded off against degree of compression.

Chapter 5. 2

This chapter and the following ones (chapter 6 and 7) will discuss algorithms that produce a reduced subset of the original data of a specified quality. The reduced subset of data points is used to generate the curve, which is an approximation of the original curve.

The present chapter will encompass real time algorithms and the next ones will deal with almost real time algorithms.

5.1 Introduction

In this chapter, we analyze point elimination algorithms which are suitable for real time; i.e it is desired to automatically and efficiently code hand generated materials in real time, so that they can be reconstructed near-instantaneously at the remote receiving location.

In recent literature in digital processing of freehand drawn material, (KEG77, DAGN79) attempts have been made to select relevant points along the trajectory of the pen. These attempts are based upon shape information; they appear not to make use of the availability of sequential and time information; their selection of important points is usually based on the distance between points. From one representative point, a subsequent point is sampled if the pen has moved a preset distance along the x or the y direction. Here the preset difference is the maximum displacement in x or y between two contiguous selected points. For further explanations, let us assume a preset distance of 2. If we express the second point in terms of differences with respect to the first, the result will be an element of the following 16 elements set:

{ (-1,2) , (0, 2) , (1, 2), (2, 2), (2, 1), (2, 0), (2,-1), (2,-2) ,
(1,-2) , (0,-2) , (-1,-2), (-2,-2), (-2,-1), (-2,0), (-2,1), (-2,2) }

WEBER65, MED65, PRY70 mention that this method was widely used in the sixties for telemetry signals which are one dimensional signals, i.e waveforms whose data are given in the form of a single-valued function

$s_i = f(t_i)$, with monotonic parameter t_i, i.e $t_i > t_{i-1}$ for (i= 0, 1, 2..).

The method has been called zero order predictor fixed aperture and is well described in the cited references.

In this present work, we have found Kegel and Dagnelie's techniques unsatisfactory for the following reasons:

1. They over represents straight lines; the ideal algorithm should sample only two points; i.e starting and finishing points.
2. They fail when the pen moves rapidly on the writing surface; in other words, material generated by a fast writer may be unacceptably distorted. As we shall see shortly, this is due to the fact that the aperture is fixed, and consequently may not faithfully cater for parts of the picture, which were generated at high pen displacement speeds.

Let us see how Kegel and Dagnelie's techniques perform on the original picture depicted in Fig.5.1; which forms part of a recorded tutorial with added text. A preset distance of 2 in x or y direction results in Fig.5.2. Compared with the original picture (Fig.5.1), we can see that the resulting distortion on some parts of the picture is unacceptable, because it is almost impossible to recover the originally intended message from the approximated picture, for example the portion of the picture which depicts a current source shows an unacceptable distortion, though most of the handwriting conveys the intended message. In this present example, the Kegel et al method has failed. The failure is due to the fact that the movement of the writing pen was very fast and distances (in x or y direction) covered by the pen in the particular sampling periods were greater than the preset distance of 2; so we can claim that Dagnelie and Kegel method does not adapt with the writing speed. It would have been better if

the preset distance was a function of the writing speed.

In 1960 A. Rényi said:

"The fact that information can be measured is by now generally accepted

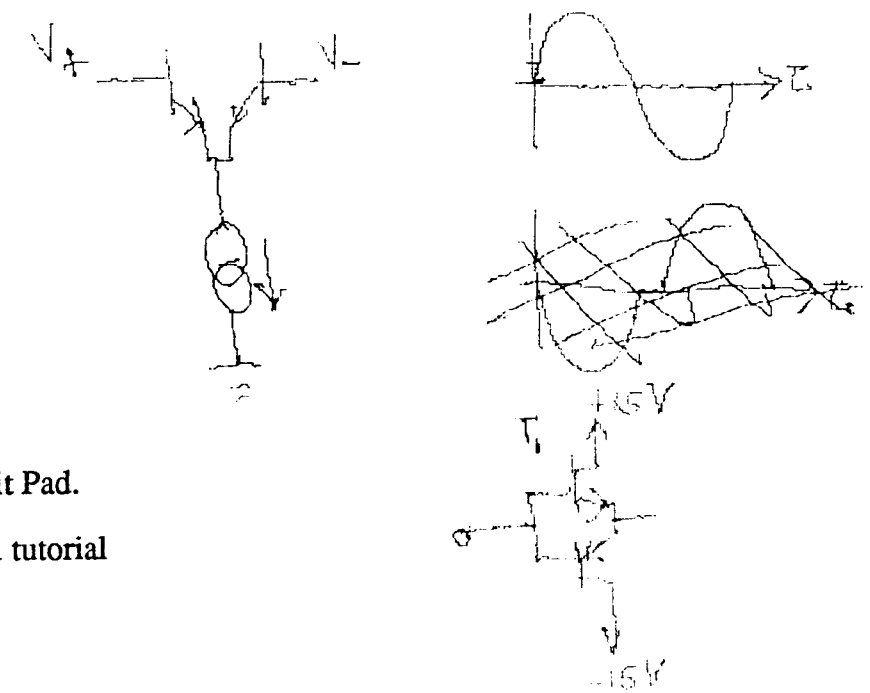


Fig.5.1

Original picture as output by The Bit Pad.

This picture forms part of a recorded tutorial with added text

In 1960 A. Renyi said:

"The fact that information can be measured is by now generally accepted

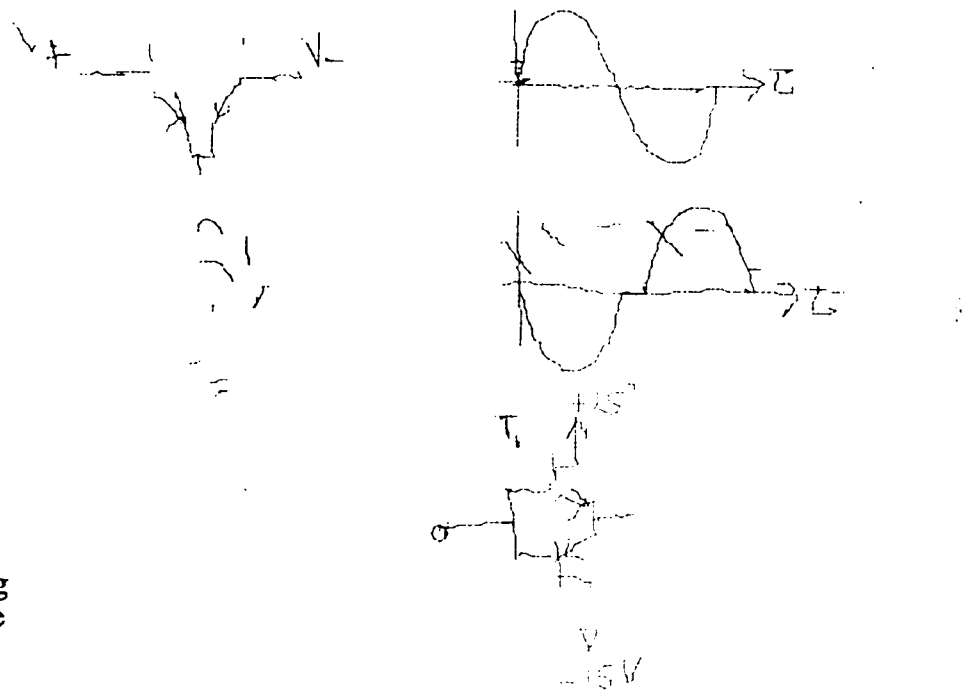


Fig.5.2
Handwriting and drawing after being segmented by a Z.O.F fixed aperture
Aperture in x direction = 2
Aperture in y direction = 2

The Kegel et al method works for slow drawing and writing, because the generated successive distances between pen positions are close and dense, this is verified in Fig.5.2, where , in this instance most of the handwriting is faithful; but the method is clearly inadequate for 'real' handwriting; Human beings have individual styles when using a pen. For example, they vary their writing speed.

In this chapter, we shall deal with three algorithms. The first one is a modification of the above technique so that it can cope with the writing speed. The two others are sufficiently distinctive to be claimed as original.

The three algorithms have been tested with two types of data. One type is generated analytically and the other was recorded in tutorials, as explained in chapter two. For evaluative purposes, the performance of each algorithm has been tested both on slow and fast freehand generated material of which the original is depicted in Fig.5.1.

As in the previous chapter, for the best algorithm, a statistical analysis will be carried out. The work described in this chapter is summarized in Fig.5.3

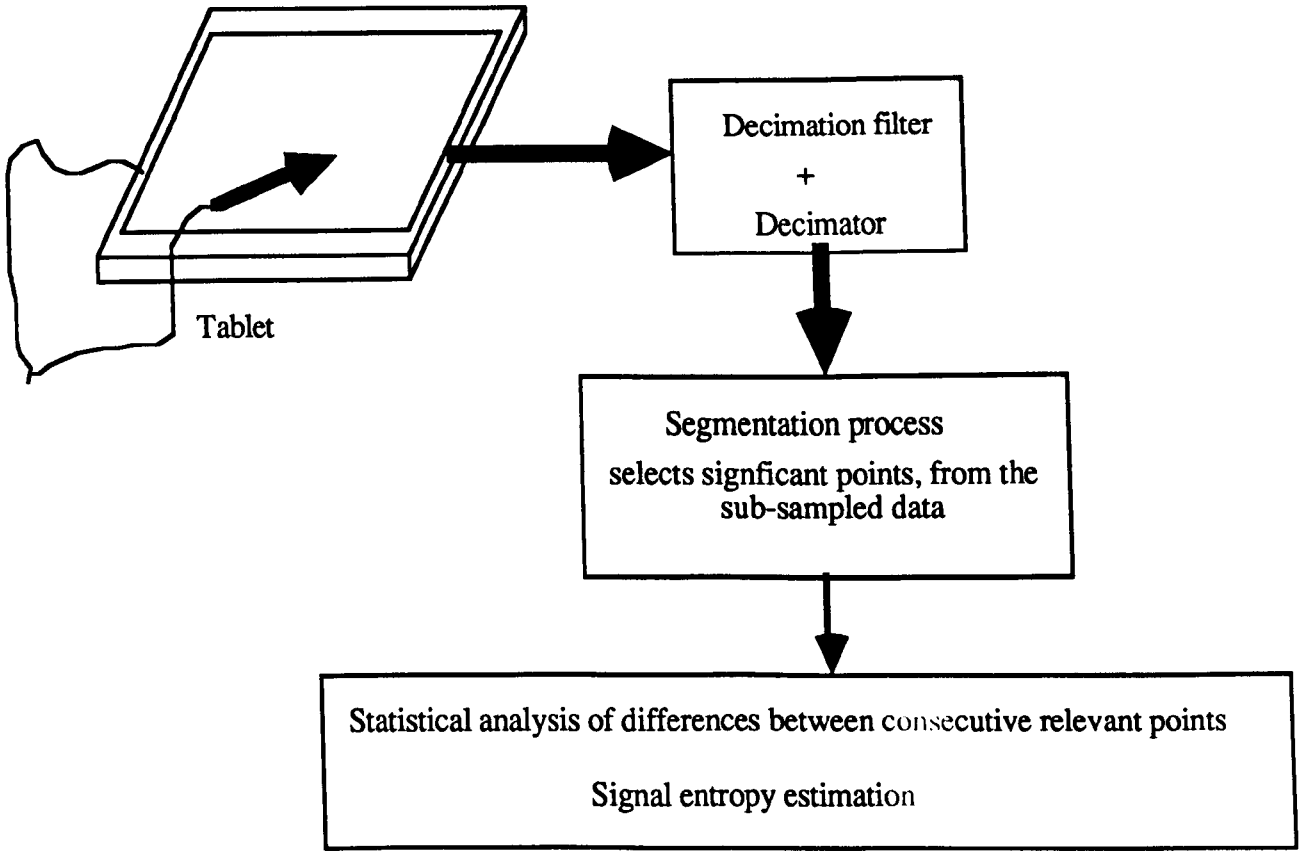


Fig.5.3

The remaining part of this chapter is organized as follows :

1. Presentation of each algorithm, analysis and performance.
2. Comparison of the three algorithms
3. Statistical analysis of the successive relevant data point differences produced by the best algorithm. The probability estimates are used to calculate an estimate of the signal entropy.

The algorithms are presented in the next section. The first two are the author's own ideas. The third is an improvement of the second algorithm. The improvements were suggested by G. READ of the Faculty of Mathematics, Open University. As in the last chapter, some points are selected and joined by straight lines. In so far as our application is concerned, legibility is an overriding requirement, so as long as the intended message is conveyed to the recipient, linear approximation may be good

enough; however if the material processed is of artistic nature, linear approximation may be rejected because it does not cater for the smoothness of the curved regions of a trace. Moreover if the approximated material is magnified, kinks will be obvious at the joints, and may not be acceptable where a smooth transition is required. This particular problem is usually attacked by using a type of curves called splines. In chapter 7, two particular splining techniques called Bezier curves and uniform B-splines will be shown to be suitable in instances where smoothness is required in hand generated material.

5.2. Algorithms.

Three algorithms for real time scan-along polygonal approximation are described. We refer to these algorithms as RTSAPA-1, RTSAPA-2, RTSAPA-3.

To describe polygonal approximations and associated algorithms, the following notation will be used. The original data are passed through a 7 to 1 decimator as described in chapter 4; the data output of the decimator are to be approximated by our algorithms. Let the decimator outputs be described by the discrete sequence P(i) (i.e x(i), y(i)) where i is the sample integer. The approximations are described by a subset of of the time sub-sampled data in which a number of the uniformly time sub-sampled data have been discarded. After decimation, if a trajectory of the pen consists of the (sample number , amplitude) pairs

$$\{(i, P(i)), i = 0, 1, 4, ...\}$$

the approximation is represented by the reduced data

$$\{(i, P(i), \ i = r_0, r_1, r_2.....\}$$

The (sample number, amplitude) pairs of the reduced data are termed relevant points of the uniformly decimated original data. Between any two

relevant points, the data are approximated points on the straight line joining the two relevant points.

The line option of the graphic package, D.I.G.S, available in the Unix system has been used to generate the straight line between the relevant points produced by each algorithm. The output device was the 7225A Hewlett Packard plotter, with a very high resolution; i.e the minimum resolvable pen displacement on the platter, is 0.033 mm. The minimum resolvable pen displacement on the writing area of the input device (Bit pad) was recorded to be 0.5 mm (SMOL81). So the input device and the output device have a 16 to 1 resolution ratio. Such a high resolution for the display device, means that if any segmentation produces visually accepted pictures, it will automatically be successful for lower resolution display devices.

The performance of data reduction will be evaluated in terms of data point reduction rate and compression ratio factor, respectively defined as :

$$d.p.r.r = (1 - NSP/TNP)*100\ \% \qquad (5.1)$$

and

$$C.R = TNP / NSP \qquad (5.2)$$

where TNP, NSP are respectively the total number of points and the number of selected points.

On all approximated analytical curves, the squared marks represent the relevant points. D.I.G.S has the ability of interpolating with a straight line between the relevant points. The objective tests were conducted on three analytical curves; a straight line, a logarithmic spiral and an astroid. The equations used for the generations are in appendix ACHAPTER5. We now develop and analyze the algorithms.

5.2.1 RTSAPA-1 (floating aperture predictor method)

This algorithm is just a modification of methods published by Dagnelie and Kegel. As indicated above, their algorithm may produce an unacceptable shape distortion, on material which was generated by fast human writers. For coping with the fast displacement of the pen on the digitizing tablet, a simple spatial filtering is performed as follows :

Let u(i) represent either the horizontal coordinate x(i) or vertical coordinate y(i); let i = r denote the sample number of the most recently determined relevant point. When beginning the approximation algorithm, the starting point P(0) = (x(0) , y(0)) of a pen trace is always declared relevant. Assuming that the sample u(0) represents either abscissa x(0) or ordinate y(0), we place two boundaries, defining an aperture at

u(0) + K and u(0) - K. The quantity K is the prediction tolerance. As long as subsequent sample values u(1), u(2) ....fall within the aperture, no action is taken. The first time, say at i = r, that a sample falls out of the aperture or is on the boundaries, u(r) is selected as a relevant point; the aperture boundaries are moved to

u(r) + K, u(r) - K, and the process is continued. The algorithm works on both directions (x and y); whenever the test succeeds in a particular direction, the corresponding (x(r), y(r)) is selected as a relevant point. This technique has been called a zero order predictor with floating - aperture (ANDREWS67) . In his paper, ANDREWS67 deals with the method in the context of a single valued function. In this research the data may represent multivalued functions (eg a curve can turn back on itself).

The principle of the floating aperture predictor technique is shown in the following diagram, which is followed by the flow-chart (RTSAPA-1) of the computer program for zero order predictor with floating aperture.

The symbols used in the flow-chart (RTSAPA-1) are:

i; index of the last transmitted sample. IMAX; total number of samples.

j; index of the sample to be processed.

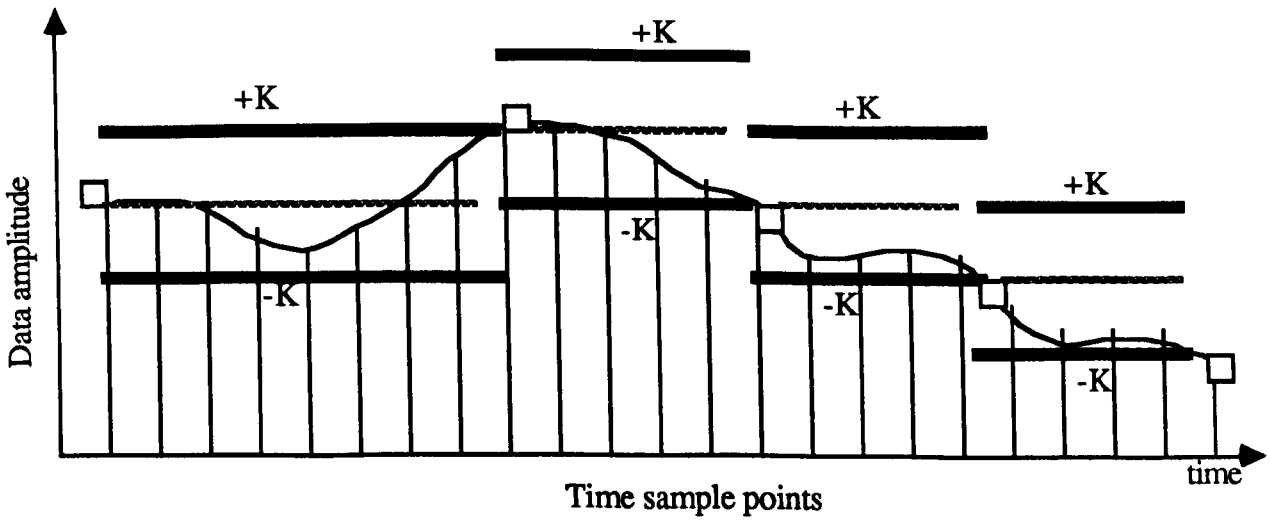
k; number of redundant samples.

C.R ; compression ratio. Tol is the tolerance K.

The following diagram indicates that, a zero order floating-aperture technique, selects a data sample only if it differs from the predicted value by an amount equal or greater than the specified tolerance limits (i.e +K, -K).

An aperture of -+K is placed about the last relevant data point. If the following data point lies within this aperture, it is not selected and the aperture is placed about it (repeating the process). If $P(t_n)$ is not selected, $P(t_{n+1})$ is compared again with the last selected sample -+K and so on.

Relatively few samples are significant and need be transmitted to convey the data behaviour.



Relevant point

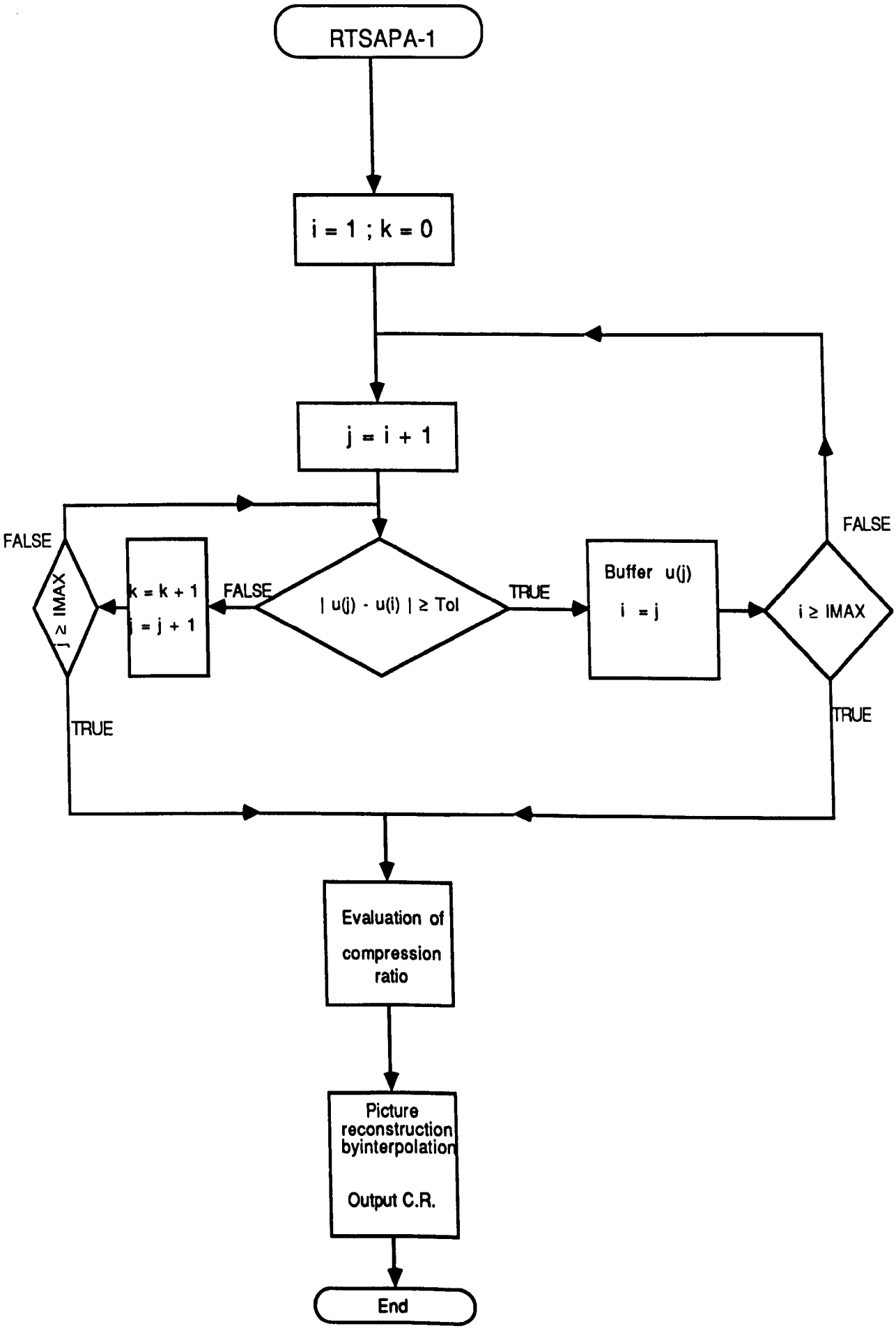
Predicted values

Data behaviour

+K upper tolerance limit

-K lower tolerance limit

Sample points


RTSAPA-1
i = 1 ; k = 0
j = i + 1
| u(j) - u(i) | ≥ Tol
FALSE
TRUE
k = k + 1
j = j + 1
j ≥ IMAX
FALSE
TRUE
Buffer u(j)
i = j
i ≥ IMAX
FALSE
TRUE
Evaluation of compression ratio
Picture reconstruction byinterpolation
Output C.R.
End

The floating aperture slightly differs from the fixed aperture predictor used by Dagnelie, Kegel.

We consider the zero order predictor floating point aperture as a simple spatial filter; we express the above in two dimensions as follows:

Essentially, the filter compares the absolute value of differences between the last accepted point and the current point; if either absolute value of Δx or Δy is greater than or equal to the spatial filter constant, spfc, the point is accepted. The filtering process is started at the first point of a trace. Different filter constants can be applied to the horizontal and vertical directions and could be formulated as follows :

Accept the incoming co-ordinate (xp, yp) only

$$\text{if } |\Delta x| = |x_p - x_l| \geq \text{spfcx}$$

$$\text{or } |\Delta y| = |y_p - y_l| >= \text{spfcy}$$

p and l stand respectively for the present co-ordinate under test, and the last co-ordinate which has been already pronounced significant, and therefore relevant; and spfcx, spfcy are respectively the spatial filter constants in the horizontal and vertical directions.

The filter constant defines a window (i.e an aperture), the ultimate aim of our window search is that the original picture must be reconstructed with a guaranteed fidelity.

Error estimates

As long as the data capture rate is high enough, the original signal does not go out of the aperture and return within it during the interval between two samples, the interpolation error will be less than |spfcx| in the x direction

and less than |spfcy| in the y direction for all sample intervals except the last.

Analysis of the experimental results

Fig 5.4 shows the straight line processed by RTSAPA-1; the spatial filter constant in x or y direction is 10. From 175 original points derived analytically, 25 were output by a 7 to 1 decimating process, and only 11 were selected by RTSAPA-1. With respect to the decimating process , we get a compression ratio of 2.27, this corresponds to 56 % data point reduction ratio. Although 56 % of points have been rejected, we can still see that the straight line is over represented; an ideal algorithm should sample only the two end points. An overall compression ratio is 15.9, in other words 93.7 % of the original points were discarded.

When the filter constant in either direction is less than or equal to 10, the approximation of the logarithmic spiral remains acceptable ; this is verified in Fig.5.5. From 1090 original points, 45 points were selected, this is a 3 to 1 compression with respect to the decimating process.

The starting region of the spiral is heavily distorted for filter constants greater than 10; this is illustrated in Fig.5.6, where the filter constant in both directions show is 16. It is clear that RTSAPA-1 cannot cope well with the regions of high curvature.

After the trials on data derived from mathematical curves, RTSAPA-1 was tested on real experimental data generated from the digitising tablet. A series of trials on 13 tutorials has shown that in terms of visual requirements, compressed handwriting generally requires a much higher number of significant points to produce an acceptable picture. So our optimum aperture has been reached on the basis of pages of mainly handwritten texts. The filter constants $0 \leq spfcx \leq 3$ and $0 \leq spfcy \leq 3$ have produced a visually acceptable picture; they work for both handwriting

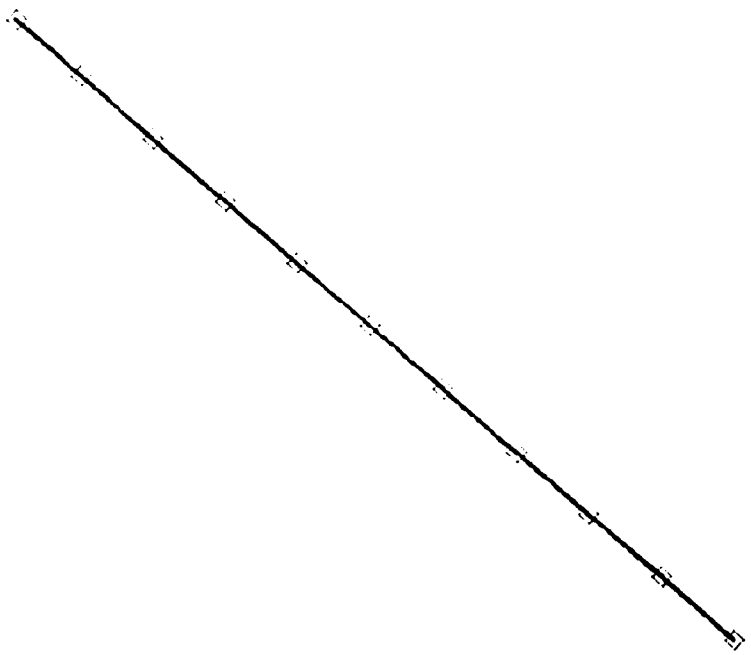


Fig.5.4
RTSAPA-1 segmentation of
a straigtline
filtert constant : spfcx = 10; spfcy = 10.
□ Indicates the selected points.

Fig.5.5
RTSAPA-1 segmentation of
Archimedes Spiral
Filter constant : spfcx = 10; spfcy = 10

□ Indicates the selected points.

Reconstruction using a straight line`
interpolator

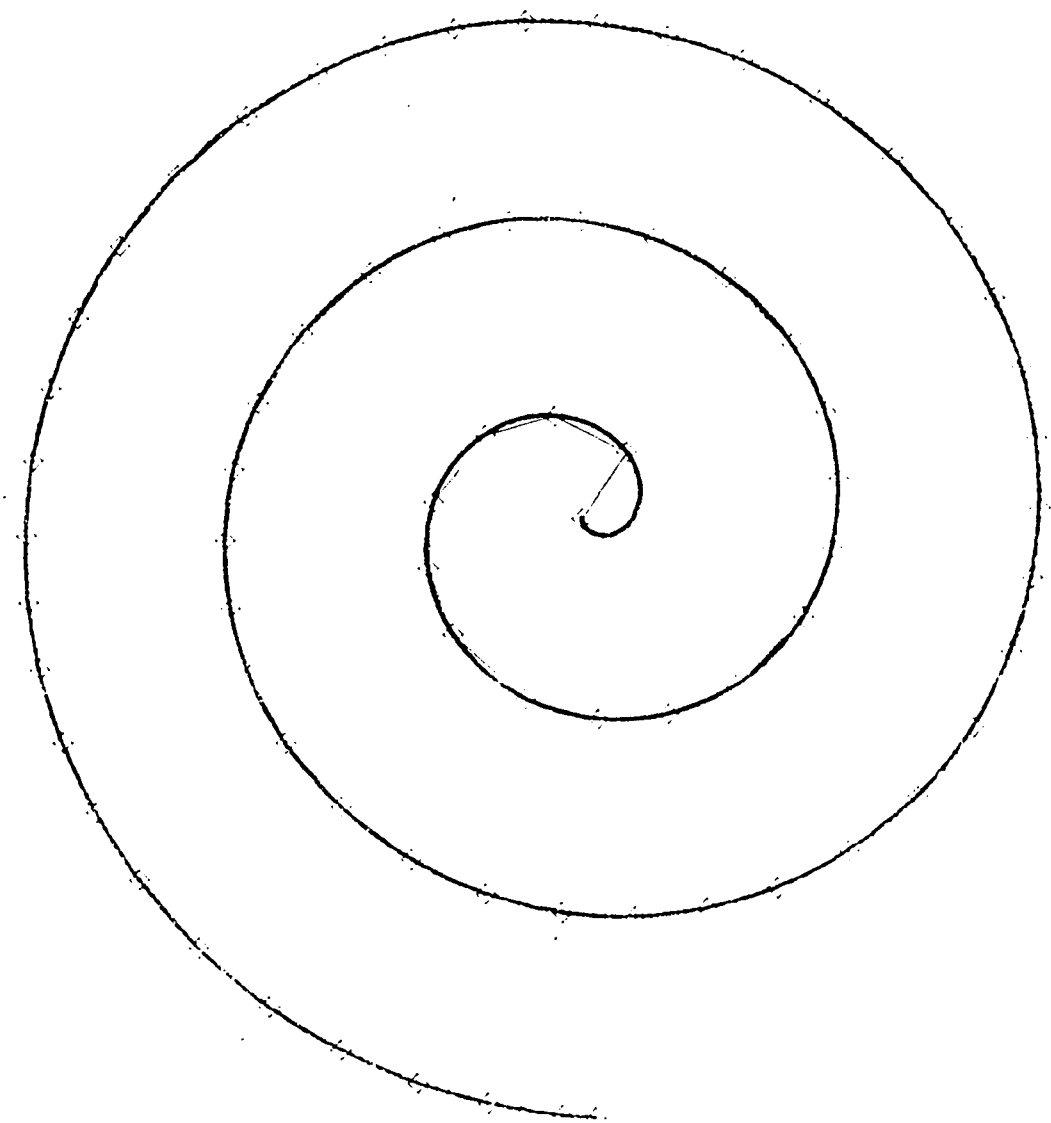
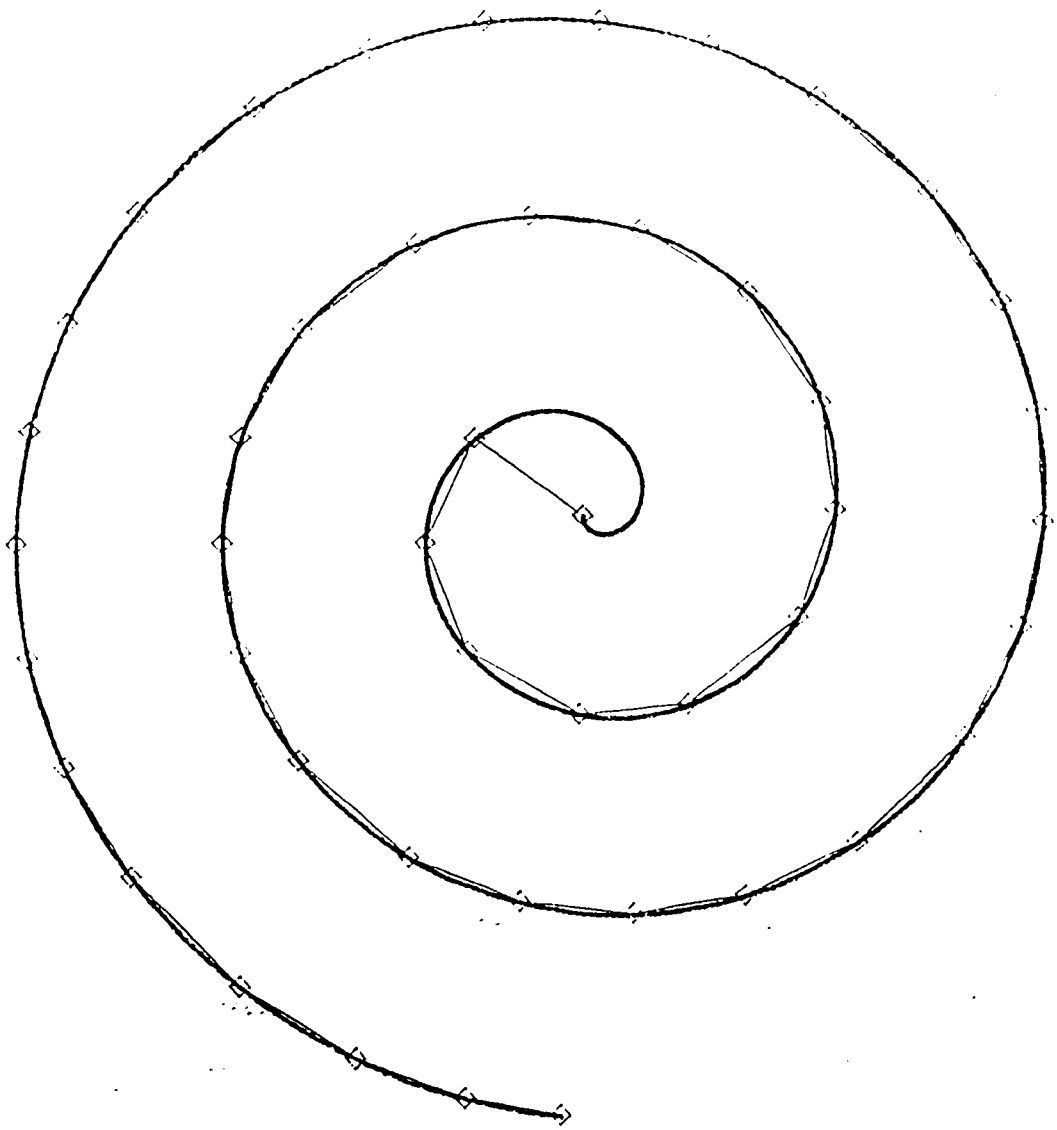


Fig.5.6
RTSAPA-1 segmentation of
Archimedes spiral
spfcx = 16; spfcy = 16
□ Indicates the selected points.



and drawing. Fig.5.9 shows the approximation for the case of spfcx = 3, spfcy = 3. This amounts to rejecting all points in the area depicted in Fig.5.8 after one relevant point has been accepted. In other words the area defines a prohibited (i.e exclusion) zone. All points external to the prohibited zone are candidates for the acceptance. The prohibited zone is defined by $|\Delta x| <$ spfcx and $|\Delta y| <$ spfcy, where $(\Delta x, \Delta y)$ represents the move between the last selected point and the incoming point.

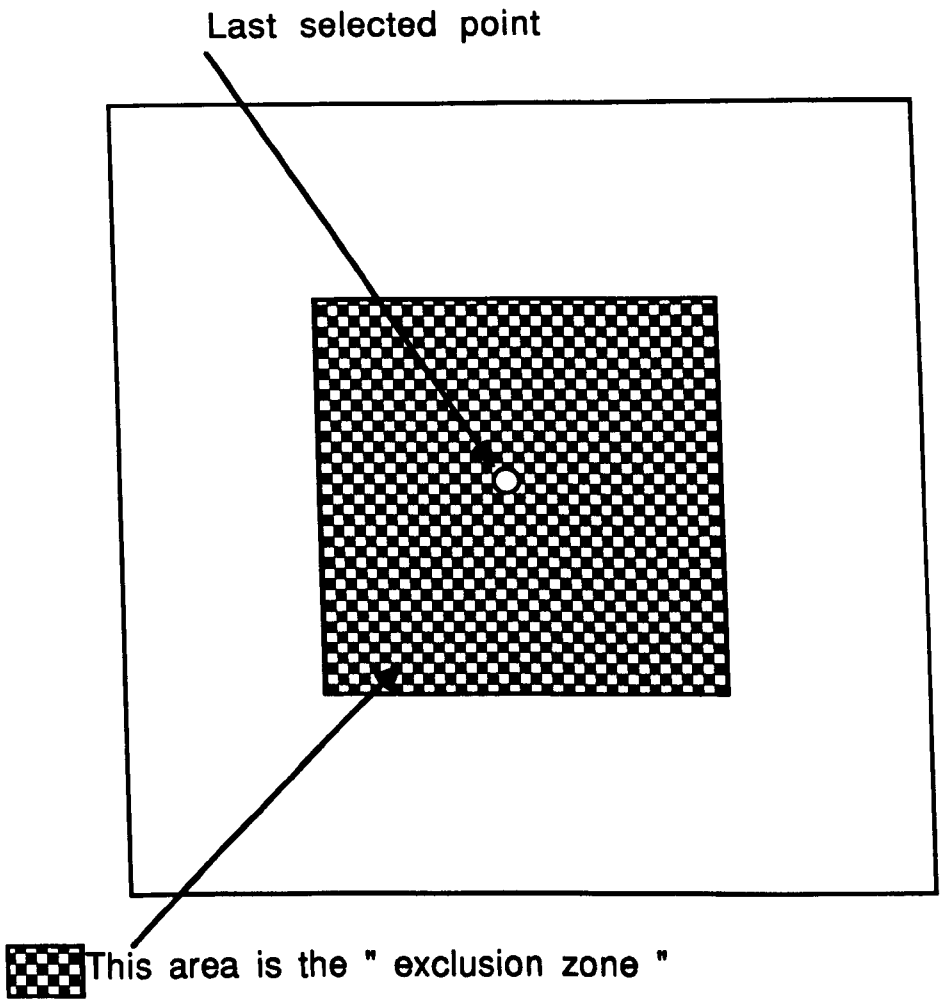


Fig.5.8

In terms of visual perception, no significant change seems to have occurred, there may be a slight deterioration of the handwritten part of the picture; but it is quite difficult to detect.

The sub-sampled picture is shown in Fig.5.7; for a 7 to 1 decimation, it requires 4348 points. Applying RTSAPA-1 to the sub-sampled data

produces 1638 points; so with respect to the decimator, we get a compression ratio of 2.65. So 62 % of sub-sampled data were rejected by the floating aperture filtering technique.

With respect to the number of points (i.e 21736), describing the original picture, we get a 13.27 overall compression ratio - an overall point reduction rate of 92.46 %.

For filter constants greater than or equal to 4, the deterioration of the handwriting is obvious: some letters are literally missed out. However the drawing is still acceptable. This is illustrated in Fig.5.10. A great deal of experiments on handwritten texts were carried out, and led us to discover that a filter constant of 4 is the critical value for handwriting.

Working on 13 tutorials, produced by different tutors, our experience shows the drawing may be still acceptable, as long as the filter constant in both direction is less than 13, this is illustrated in Fig.5.11. Looking at the drawn part of the picture displayed on Fig.5.11, a well trained electronic engineer may still recognize the current source; but the written information is completely lost. On the basis of trials on substantial data sets, we have found that a spatial filter characterised by spfcx = 3 and spfcy = 3 eliminated significantly highly variable, non essential details; this means that, given 0.5 mm writing surface resolution (SMOL81), and owing to the Nyquist sampling rate of 29 Hz (see previous chapter on decimation) a new point is pronounced significant whenever the x or y co-ordinate changes 1.5 mm form the previous significant point.

An obvious weakness of the zero order floating aperture is that its operation is not independent of a scale factor, presumably this implies that, if the coordinates of the points are scaled by a factor K, to get the same pictorial results our filter constant would have to be multiplied by K.

In 1960 A. Rényi said:

"The fact that information can be measured by any generally accepted

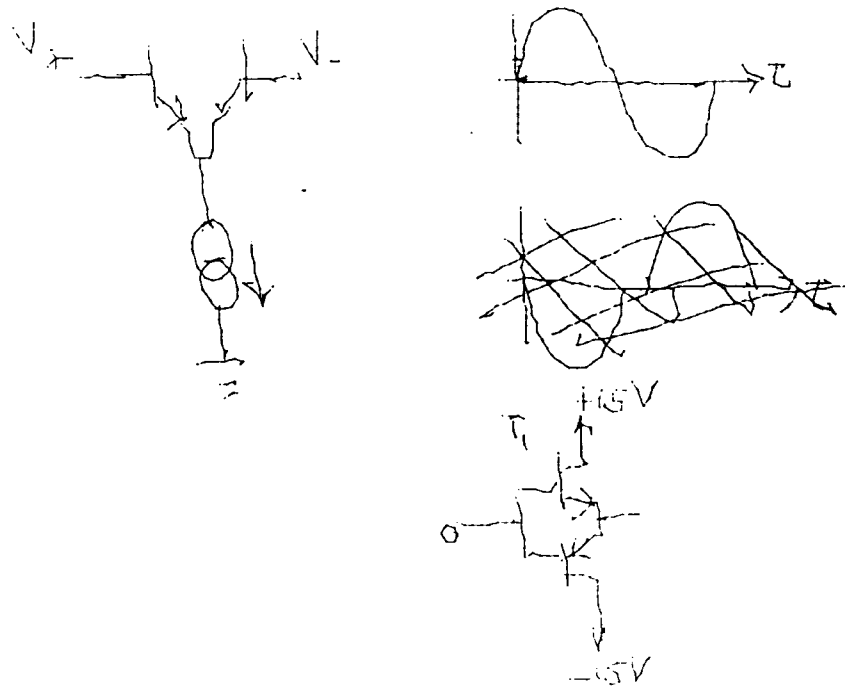


Fig.5.7
Straight line reconstruction of handwriting and drawing after passing through a 7 to 1 decimating process

In 1960 A. Renyi said:

"The fact that information can be

measured to any most generally accepted

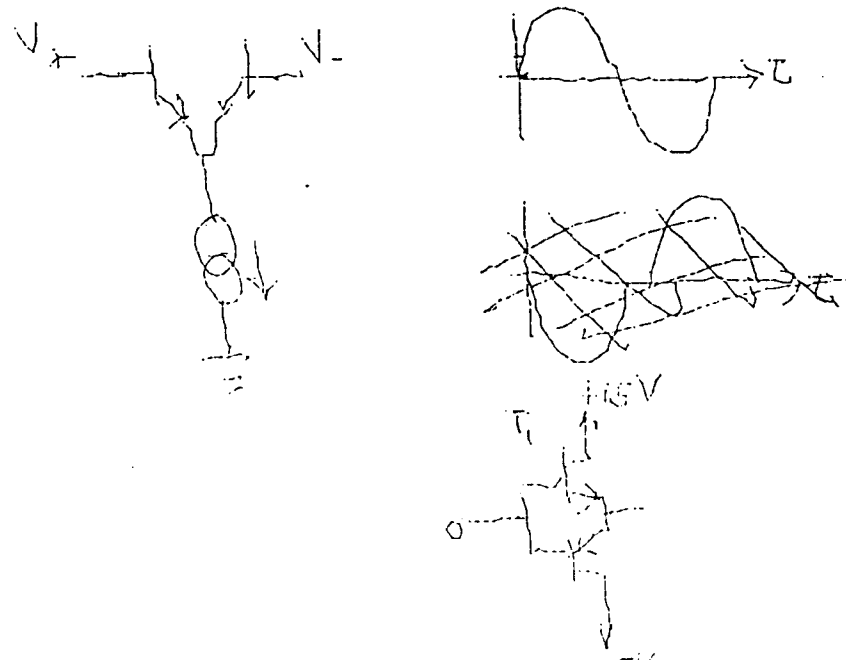


Fig.5.9
Reconstructed handwriting and drawing after being segmented by RTSAPA-1 Filter constants: spfcx = 3; spfcy = 3.

produces 1638 points; so with respect to the decimator, we get a compression ratio of 2.65. So 62 % of sub-sampled data were rejected by the floating aperture filtering technique.

With respect to the number of points (i.e 21736), describing the original picture, we get a 13.27 overall compression ratio - an overall point reduction rate of 92.46 %.

For filter constants greater than or equal to 4, the deterioration of the handwriting is obvious: some letters are literally missed out. However the drawing is still acceptable. This is illustrated in Fig.5.10. A great deal of experiments on handwritten texts were carried out, and led us to discover that a filter constant of 4 is the critical value for handwriting.

DUPLICATE

Working on 13 tutorials, produced by different tutors, our experience shows the drawing may be still acceptable, as long as the filter constant in both direction is less than 13, this is illustrated in Fig.5.11. Looking at the drawn part of the picture displayed on Fig.5.11, a well trained electronic engineer may still recognize the current source; but the written information is completely lost. On the basis of trials on substantial data sets, we have found that a spatial filter characterised by spfcx = 3 and spfcy = 3 eliminated significantly highly variable, non essential details; this means that, given 0.5 mm writing surface resolution (SMOL81), and owing to the Nyquist sampling rate of 29 cps (see previous chapter on decimation) a new point is pronounced significant whenever the x or y co-ordinate changes 1.5 mm form the previous significant point.

An obvious weakness of the zero order floating aperture is that its operation is not independent of a scale factor, presumably this implies that, if the coordinates of the points are scaled by a factor K, to get the same pictorial results our filter constant would have to be multiplied by K.

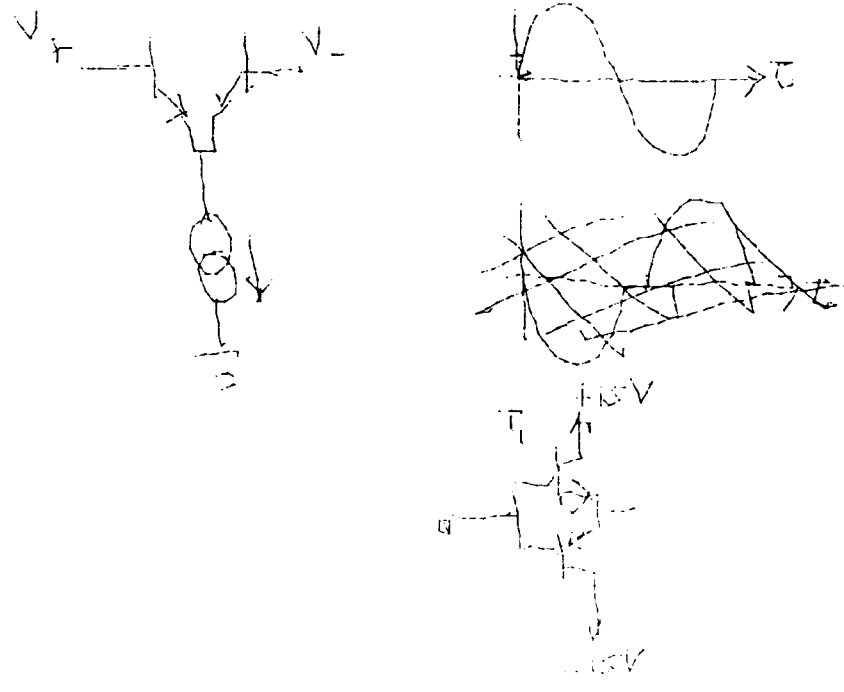


Fig.5.10
Reconstructed handwriting and drawing after being segmented by RTSAPA-1
Filter constants: spfcx = 4; spfcy = 4.

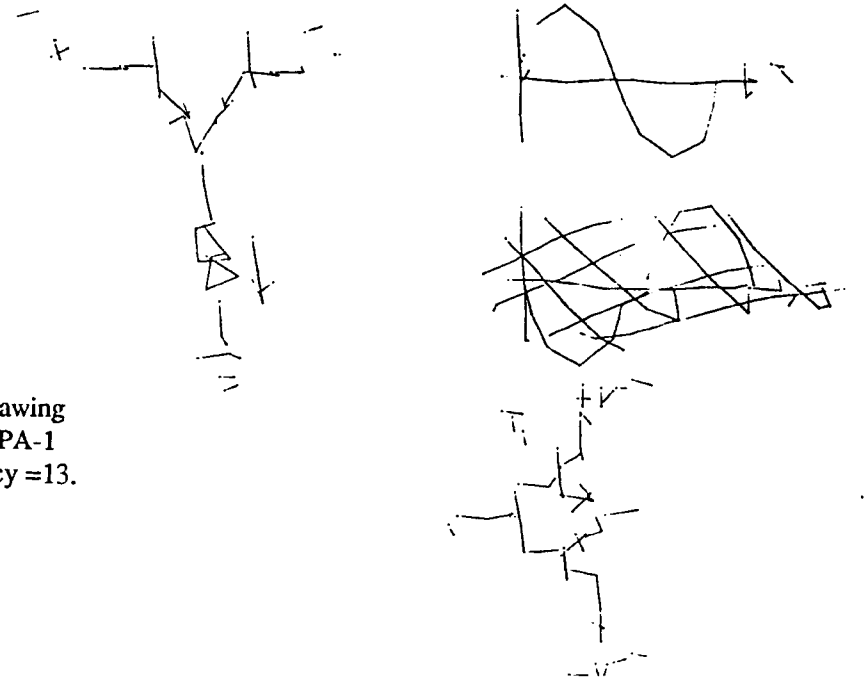


Fig.5.11
Reconstructed handwriting and drawing
after being segmented by RTSAPA-1
Filter constants: spfcx = 13; spfcy =13.

5.2.1.1 RTSAPA-1 Evaluation based on accuracy criteria.

The impairment of the reconstructed pictures grows with the spatial filter constant. The visual accuracy measure is inversely proportional to the spatial filter constant. Thus the higher the filter constant, the lower the visual accuracy. So an acceptable visual accuracy leads to the ultimate filter constant. In our particular application, a perceivably acceptable picture means that the recipient can still extract the intended communication from the impaired pictures satisfactorily.

5.2.1.2. RTSAPA-1 Evaluation based on efficiency criteria.

Fig 5.7 shows results from approximating a sequence of 21736 points defining the picture displayed on Fig.5.1, using spfc = 3. A comparison of processing rates has been made by coding RTSAPA-1 in C and running it on a VAX11/750 using various settings of the filter constant spfc. On average it took 0.3 seconds to process 21736 points, thus the points were processed at a rate of 72454 points per second. There is no doubt that RTSAPA-1 is very fast, this is explained by the fact that RTSAPA-1 contains no multiplications or divisions.

RTSAPA-1 is a sequential algorithm, its complexity is 0(N); N being the number of points to be processed. The operations are executed only once, and are repeated at most N times.

5.2.1.3 RTSAPA-1 Evaluation based on compactness criteria.

Table 5.1 shows the compactness performance of RTSAPA-1 for various setting of the spatial filter constant. The results displayed in Table 5.1 were obtained by feeding the whole data base to a 7 to 1 decimator followed by RTSAPA-1. At this point it is worth noting that throughout this thesis, by whole data base is meant the set of points obtained by appending together the data files which represent the tutorials recorded during Open University

Summer School 1981 (SMOL81).

The total number of data point (x,y), TNP, from the whole data base is 1283876. NSP, the number relevant points is listed in column 3 for various settings of the spatial filter constant. The compactness performance is related to the data which were obtained using the minimum acceptable sampling rate (29 cps) (see previous chapter for sub-sampling studies).

spfcx	spfcy	NSP	C.R.	D.R.R (%)
0	0	210953	1	0
1	1	183438	1.15	13
2	2	171507	1.23	19
2	3	165209	1.28	22
3	2	163315	1.3	23
3	3	126267	1.67	40

Table 5.1 Compactness performance of RTSAPA-1

The compactness performance of RTSAPA-1 can be evaluated either in terms of compression ratio, C.R. or data reduction rate, D.R.R, respectively listed in columns 4 and 5 of Table 5.1.

For a very good visual accuracy when using RTSAPA-1, 40 % of data points do not matter. This represents a good data reduction rate. The resulting data reduction rate is due to a spatial filter constant of 3 applied to vertical and horizontal directions. With respect to the whole database (1283876 points), we get a 10.17 overall compression ratio - an overall reduction rate of 90.17 %.

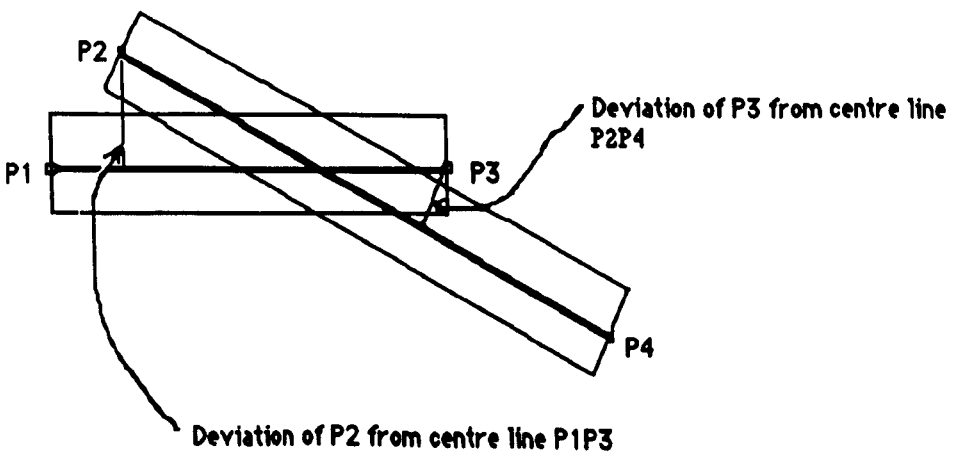
When the filter constant is 0, the total number (NSP = 210953) of relevant points is that produced by the decimator. This is the baseline for evaluating the compactness performance of our algorithms.

The obvious drawback of RTSAPA-1 is its inefficiency in taking care of straight lines (e.g Fig.5.4), so the immediate requirement of the following algorithm is to correct that deficiency.

5.2.2 RTSAPA-2

This algorithm is an improvement of RTSAPA-1, which has shown some weakness in so far as straight lines are concerned. RTSAPA-1 still over-represents straight lines. To overcome this deficiency, the output by RTSAPA-1 is processed as follows:

The selection of the relevant points from the output of RTSAPA-1 is best explained with reference to the following figure.



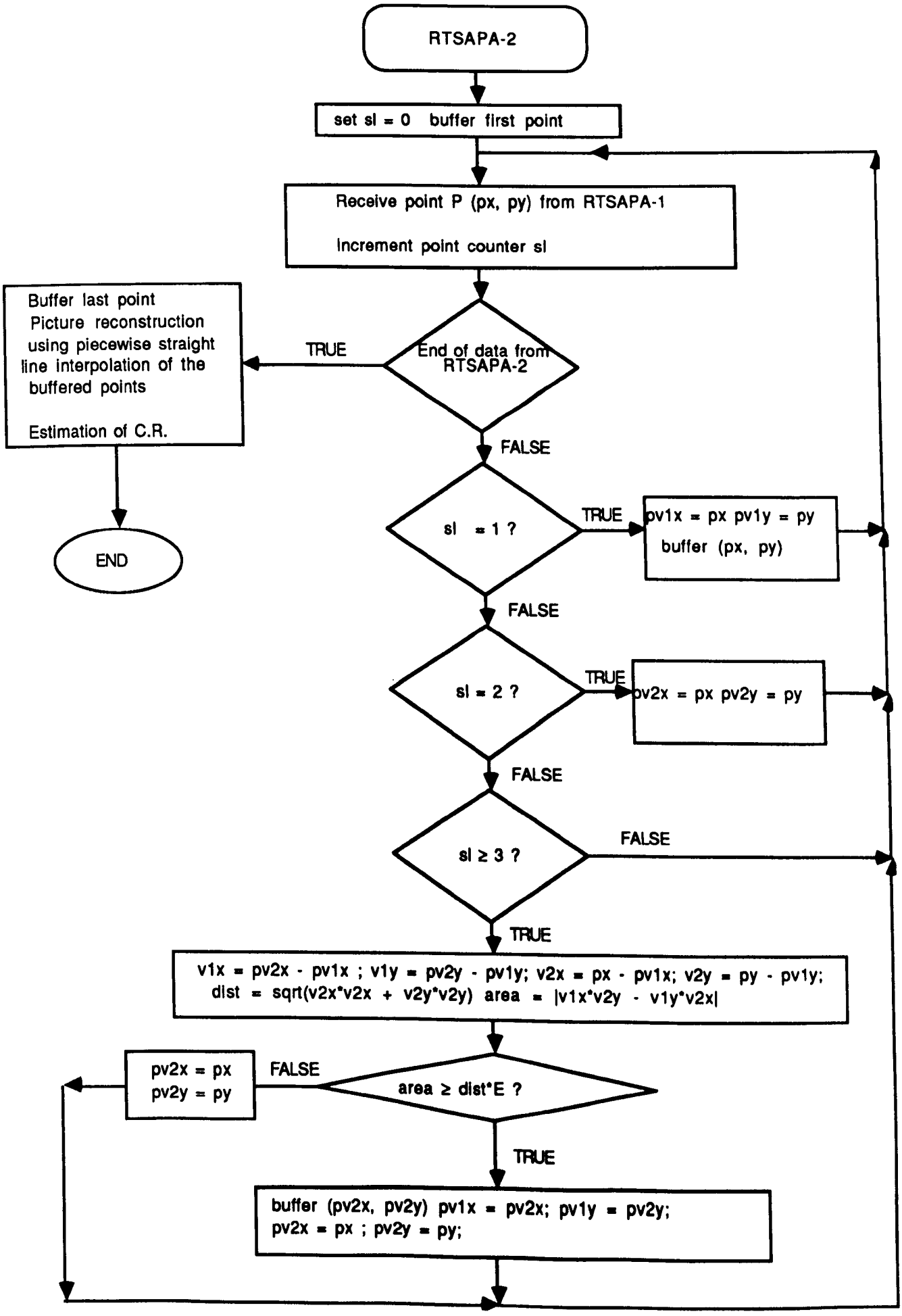
Let us say, RTSAPA-1 emits the points P_1, P_2, P_3, P_4. A mask is placed over the first three points P_1, P_2, P_3 so that the mask centerline coincides

If the intermediate point P_2 falls outside the mask (as in the above picture), then the intermediate point is selected as a relevant point, and the beginning of the mask is placed over the selected point in order to continue the processing. If the intermediate point falls within the mask, it is rejected. The point previously at the end of the mask is now taken to be the intermediate point, and the next point is taken to be at the end of the mask. The very first point and the very last point of a trace are always selected. The length of the mask is increased dynamically as needed. The width of the mask is the threshold deviation required. The above process is further explained as follows:

Let E and D, be respectively the threshold deviation and the computed deviation. We compare E with the deviation of point P_2 from the straight line defined by P_1 and P_3. The point P_2 can either be a candidate for exclusion if the deviation is less than the threshold deviation E, or for selection if the deviation is greater than or equal to E.

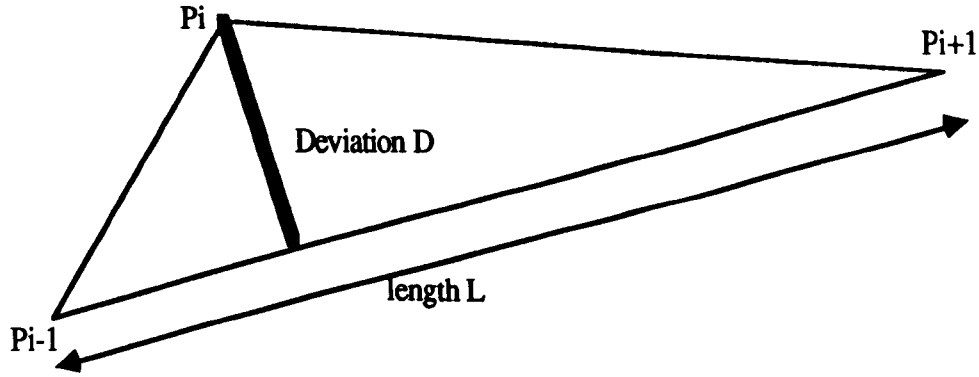
If $D \geq E$, the line segment which connects P_1 to P_2 is the first approximate line segment. If $D < E$, P_3 takes the place of P_2 we add the next point (i.e P_4) and and compute a new value of D, which is then used to decide whether the point P3 should be selected or rejected. Subsequent points are dealt with in the same way.

By this process all data points of a trace are approximated by a certain number of line segments equal to the number of selected points minus one. The flow-chart (RTSAPA-2) of the computer program for RTSAPA-2 is shown on the following figure.


RTSAPA-2
set sl = 0 buffer first point
Receive point P (px, py) from RTSAPA-1
Increment point counter sl
End of data from RTSAPA-2
TRUE
Buffer last point
Picture reconstruction using piecewise straight line interpolation of the buffered points
Estimation of C.R.
END
FALSE
sl = 1 ?
TRUE
pv1x = px pv1y = py
buffer (px, py)
FALSE
sl = 2 ?
TRUE
pv2x = px pv2y = py
FALSE
sl ≥ 3 ?
FALSE
TRUE
v1x = pv2x - pv1x ; v1y = pv2y - pv1y; v2x = px - pv1x; v2y = py - pv1y;
dist = sqrt(v2x*v2x + v2y*v2y) area = |v1x*v2y - v1y*v2x|
pv2x = px
pv2y = py
FALSE
area ≥ dist*E ?
TRUE
buffer (pv2x, pv2y) pv1x = pv2x; pv1y = pv2y;
pv2x = px ; pv2y = py;

Analysis and error estimate

The error estimate is given by the deviation from the approximate line segment. Assuming that we have three successive points P_{i-1} , P_i , P_{i+1} displayed as in the following picture :



The deviation D, can be easily found by the use of the triangle

$\Delta(P_{i-1} , P_i , P_{i+1})$

area of $\Delta(P_{i-1} , P_i , P_{i+1}) = (D^*| P_{i-1}, P_{i+1} |) / 2$

this expression equals half of the area of parallelogram

$(P_{i-1} , P_i , P_{i+1} , P_h)$, where P_h is the point of intersection of lines drawn from P_{i+1} parallel to $P_{i-1}P_i$ and from P_i parallel to $P_{i-1}P_{i+1}$. The area of the parallelogram is the absolute value of the cross product of vectors $P_{i-1}P_i$ and $P_{i-1}P_{i+1}$. The relevant expressions are :

$$D^*| P_{i-1}, P_{i+1}| = CP \qquad (5.1)$$

$| P_{i-1}, P_{i+1}|$ is the distance between points P_{i-1} and P_{i+1}. CP, the absolute value of the area of the parallelogram, is given by :

$$CP = |(y_{i+1} - y_{i-1})*(x_i - x_{i-1}) - (y_i - y_{i-1})*(x_{i+1} - x_{i-1})|$$

D is the deviation of point P_i from the line $P_{i-1}\ P_{i+1}$.

E is the threshold of the deviation D.

As $D \geq E$ for choosing P_i as relevant point, to save a division in the computer program the applied test is

$$CP \geq |P_{i-1}, P_{i+1}| E \qquad (5.2)$$

To avoid the evaluation of the square root, this test can be written

$$CP^2 \geq L^2 * E^2 \qquad (5.3)$$

where $L = |P_{i-1}, P_{i+1}|$ is the length of the distance from P_{i-1} to P_{i+1}.

The simplifications in equation (5.2) and (5.3) enable the computer program to run faster. Avoiding division and square root reduces the computing time for the algorithm.

Analysis of the experimental results

The results have been evaluated in the same spirit as those in section 5.2.1 In so far as tests on analytically generated curves are concerned, Fig.5.12 shows that RTSAPA-2 removes the deficiency of RTSAPA-1, exactly two relevant points (ie end points) are selected for a straight line. This happens because the deviations of points which build the straight line are zero, thus they never pass the test expressed in equation (5.3). The beginning and finishing points are used to reconstruct the straight line.

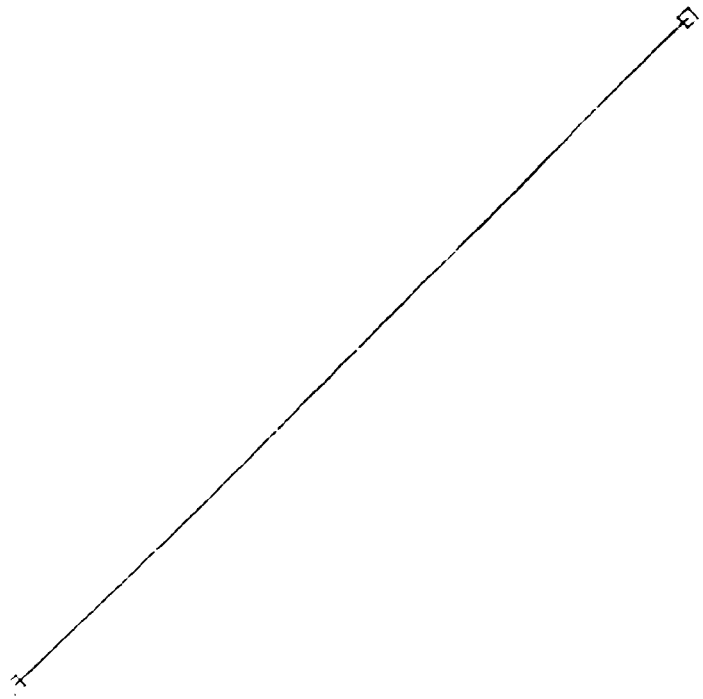


Fig.5.12
RTSAPA-2 approximation of
a straight line generated analytically.

⊓ Indicates the selected points.

The approximation of the logarithmic spiral is shown in Fig.5.13. The maximum deviation of the original points from the approximating line segments is E = 1. The approximated logarithmic spiral is visually acceptable. The approximation grows poorer when E is greater or equal to 2; this can be seen in Fig. 5.14. A careful look at this picture tells us that the ending line segment is a good approximation of the final section of the curve. This happened because the ending point is forced to be a relevant point and consequently does not undergo the test.

A question which arises at this stage is whether RTSAPA-2 can cope with the cusps which may arise from certain curves. To answer this, the algorithm was tested using an astroid.
Fig.5.15. shows that RTSAPA-2 copes very well with the cusps. The four cusps are retained during the approximation.
The maximum deviation (ie error tolerance) is E = 1.0. The general shape of the curve is preserved.

After the tests on data generated from mathematical curves, RTSAPA-2 was run with some of the experimental data produced by the digitising tablet.
Fig 5.16. shows the approximated picture for maximum deviation
E = 0.25; this approximation is visually very good, this means that the average human eye and mind will hardly make out the difference between Fig.5.16. and the original (Fig.5.1), if anything the approximated picture is smoother than the original. The approximated picture was reconstructed from 1618 points. This represents compression ratio of nearly 3 to 1, with respect to the uniformly sub-sampled data. The acceptability of the visual performance was estimated when E increased in steps of 0.25.
When E = 1, handwriting deteriorates slightly, but drawing is still very good (see Fig.5.17.), the drawing continues to be acceptable up

Fig.5.13
RTSAPA-2 approximation of Archimedes spiral. Maximun deviation from the straight line interpolator is E = 2 units of the output device (i.e Hewlet Packard plotter 7225)

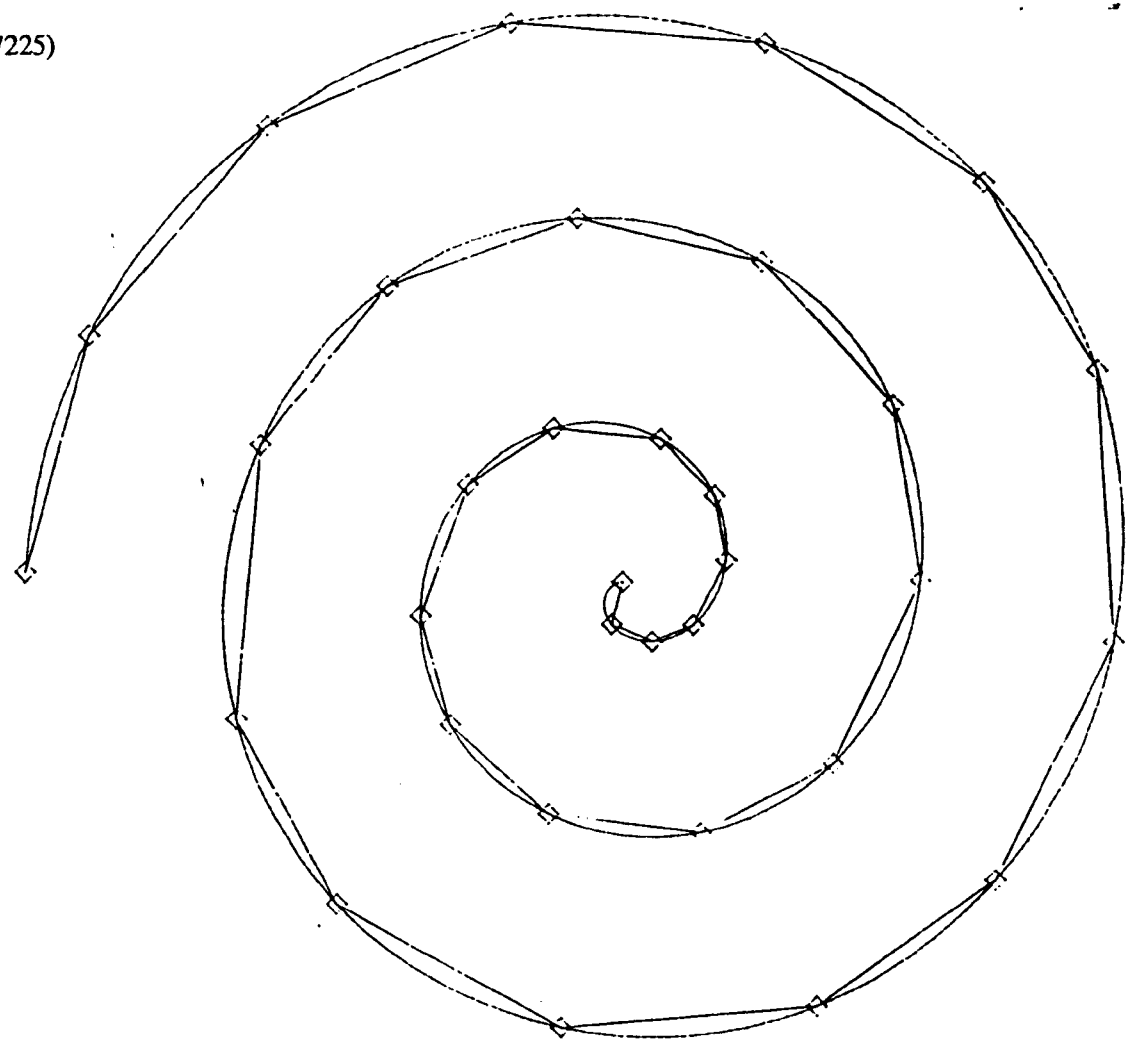


Fig.5.14
RTSAPA-2 approximation of
Archimedes spiral. Maximum deviation
from the piecewise straight line interpolator,
is E = 2 of the output device
(i.e Hewlet Packard plotter 7225)

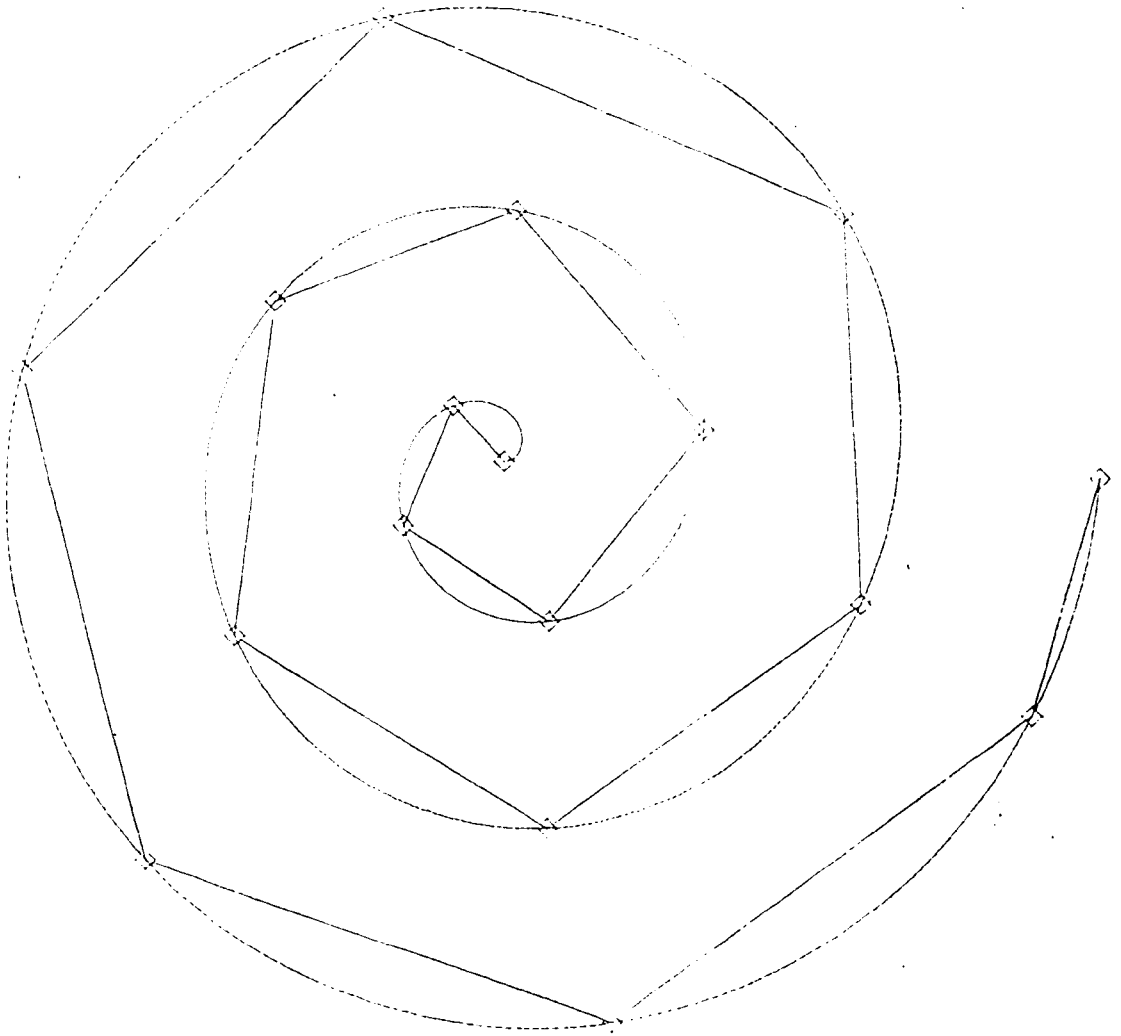
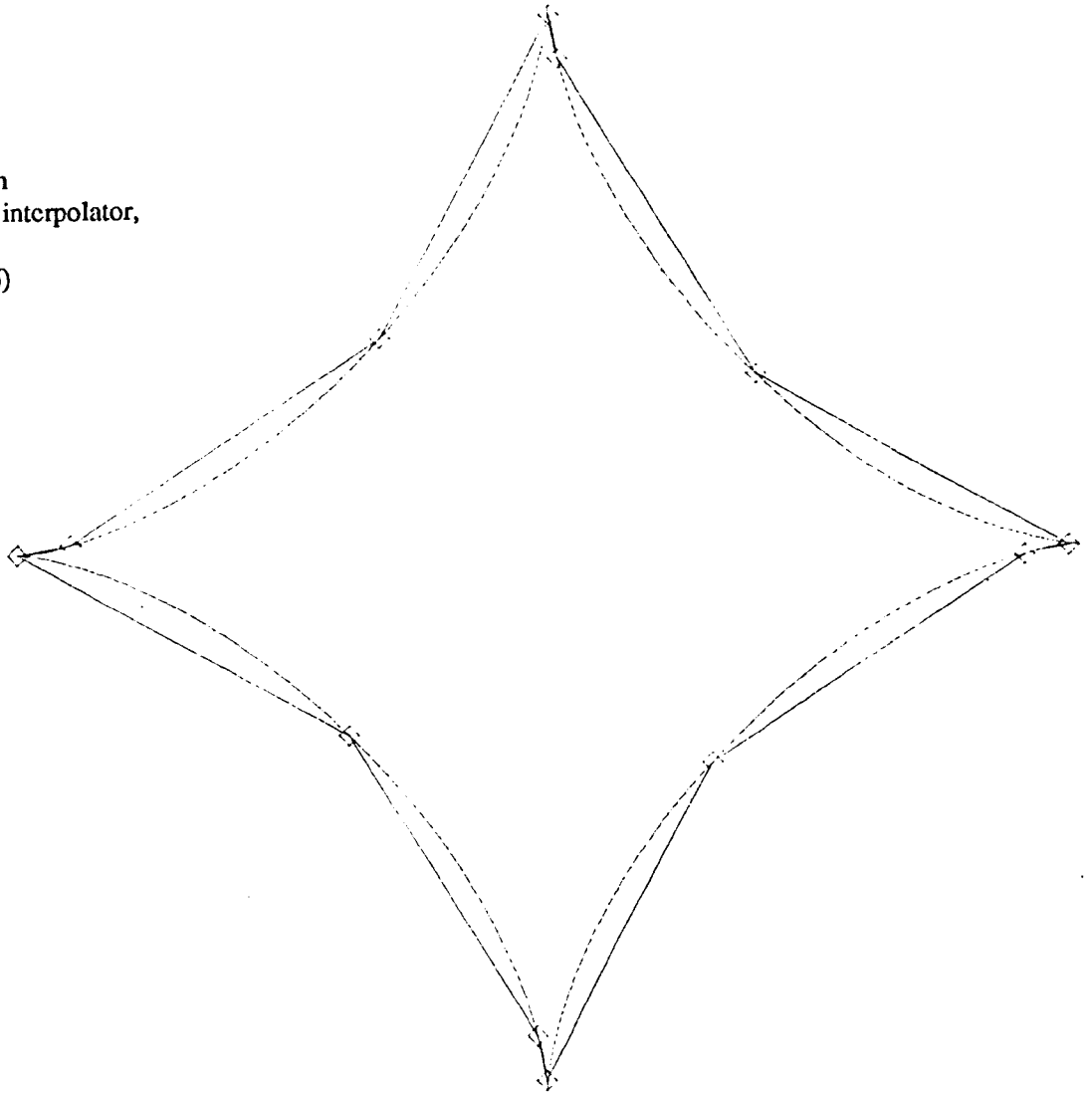
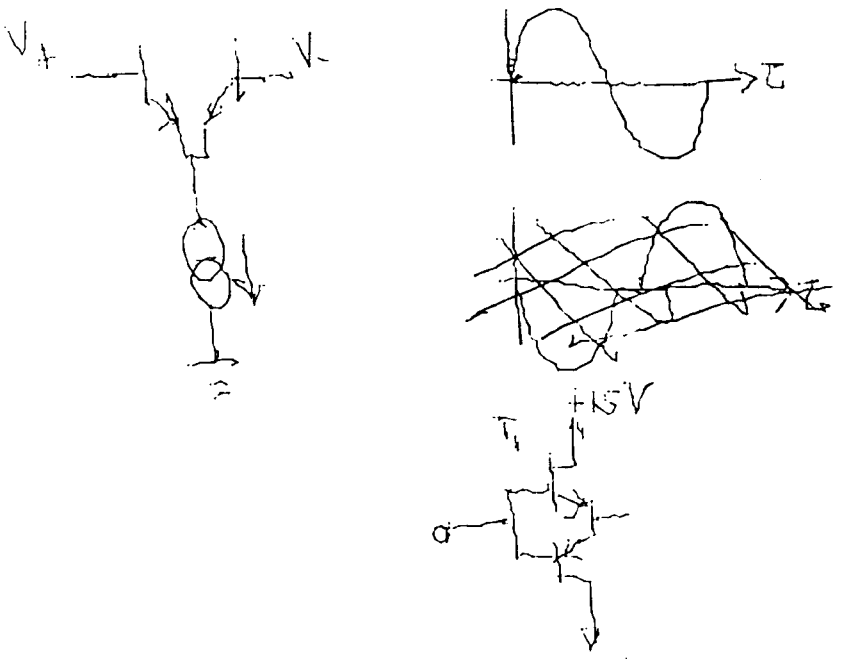


Fig.5.15
RTSAPA-2 approximation of
an astroid. Maximum deviation
from the piecewise straight line interpolator,
is E = 1 of the output device
(i.e Hewlet Packard plotter 7225)



In 1960 A. Reny said:
"The fact that information can be
measured is by now generally accepted

Fig.5.16
RTSAPA-2 approximation of handwriting and drawing. Maximum deviation from the piecewise straight line interpolator, is E = 0.25 of the output device (i.e Hewlet Packard plotter 7225)



In 1960 A. Renyi said:

"The fact that information can be measured by now generally accepted

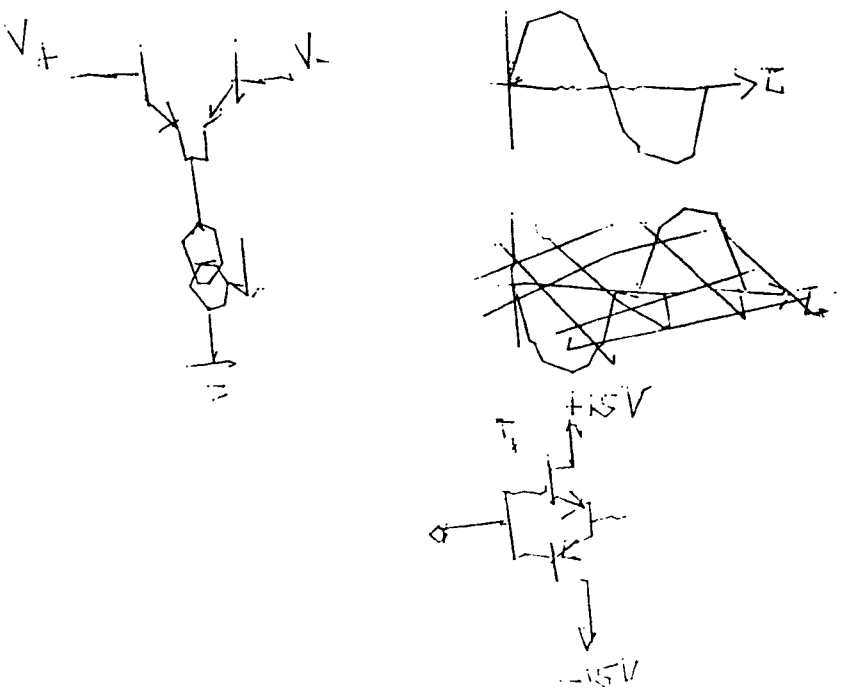


Fig.5.17
RTSAPA-2 approximation of handwriting and drawing. Maximum deviation from the piecewise straight line interpolator, is E = 1unit of the output device (i.e Hewlet Packard plotter 7225)

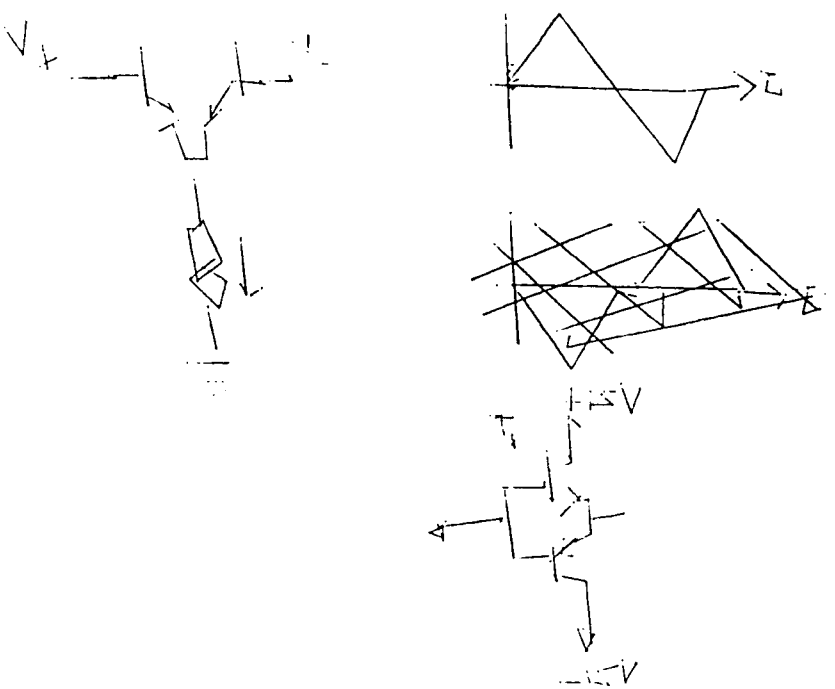


Fig.5.18
RTSAPA-2 approximation of handwriting and drawing. Maximum deviation from the piecewise straight line interpolator, is E = 1.15 unit of the output device (i.e Hewlet Packard plotter 7225)

to E = 1.150, above this accuracy tolerance, the picture is very distorted, this is illustrated in Fig. 5.18.

5.2.2.1. RTSAPA-2 Evaluation based on accuracy criteria.

The distortion of the reconstructed pictures grows with the deviation threshold. Thus the higher the threshold deviation, the lower the visual accuracy.

5.2.2.2. RTSAPA-2 Evaluation based on efficiency criteria

This algorithm requires 7 subtractions, one addition, six multiplications for the processing of each incoming point. On the VAX11/750 computer we have found that on average it takes about 1.5 seconds to process the picture Fig 5.1. which is made of 21736 points thus the points were processed at a rate of 14491 points per second. The algorithm is coded so that there is no division.

5.2.2.3. RTSAPA-2 evaluation based on compactness criteria.

Table 5.2 shows the compactness performance of RTSAPA-2 for various settings of the deviation threshold. The results displayed on Table 5.2. were obtained by feeding the whole data base to a 7 to 1 decimating process followed by RTSAPA-2.

NSP, the number of relevant points is listed in column 2 for various settings of the deviation threshold.

As in the case of RTSAPA-1 the compactness performance of RTSAPA-2 can be evaluated either in terms of compression ratio, C.R. or data reduction rate respectively listed in columns 3 and 4 of Table 5.2. For an acceptable visual accuracy, RTSAPA-2 rejects 62 % of the data points coming from the decimator.

Accuracy (E)	NSP	C.R	D.R.R (%)
0.00	210953	1	0
0.25	127174	1.67	40
0.50	91250	2.31	56
0.75	75976	2.77	64
1.00	68090	3	66
1.25	54072	3.9	74

Table 5.2 Compactness performance of RTSAPA-2

The data point reduction performance of RTSAPA-2 is good, but its accuracy performance is poor for E greater or equal to 1. Looking at equation (5.3) , with a view to improving computational efficiency, we can cut down the number of multiplications to 5 by setting the deviation to 1; but at the same time we are concerned about inspecting the visual quality of the approximated picture to see. The next algorithm will ensure that E = 1 and that the approximated picture is very good from the point of view of the visual perception.

5.2.3. RTSAPA-3

READ83 argued that in a slowly bending curve set of points, RTSAPA-2 would fail; indeed the deviation from the approximating line segment would not show that the line is bending if each deviation was compared with the threshold deviation E only. READ83 went on to suggest that the best points at which to segment a curve in order to approximate by a polygon are the points at which its direction changes abruptly ("angle") or where its radius of curvature is low. In particular where its radius of curvature is a local minimum.

READ83 proposed a method of selecting relevant points, which is a function of radius of curvature, that method is discussed in chapter 7. The following algorithm grew from READ83's ideas.

RTSAPA-3 is an extension RTSAPA-1; our concern is to reduce the number of sub-sampled points (i.e output from the decimator) that must be transmitted for effectively displaying handwritten shapes. This is accomplished by filtering out superfluous points which are generated when the pen motion is erratic, very slow or has no significant curvature or other change of direction.

RTSAPA-3 is a filtering scheme, which utilizes three independent selection criteria to eliminate any sub-sampling point that is :

a. Too close to the point at which the pen was placed upon the paper to begin writing.

b. Displaced by less than a predetermined minimum distance from the previously retained point. (This is RATSAPA-1)

c. Or has less than a predetermined angular displacement from the direction in which the pen was travelling when the previously retained point was sampled.

We have just spelled out a sequence of three filtering operations; the sequence in which these filtering operations is performed is described below.

The "pen down " filter eliminates points representing unwanted excursions of the pen that may result from erratic hand motions in positioning the pen upon the paper or in overcoming the initial friction when writing begins. The vertical and horizontal displacements Δx and Δy between the pen down

point and the succeeding points are compared to a desired threshold displacement; When a point occurs with a displacement exceeding this threshold, the pen down filter is turned off, and the point is retained. This is effectively RATSAP-1.

The filter for testing the distance from the previously retained point detects abnormally slow pen motion, and it becomes effective after the pen down threshold condition is satisfied. The horizontal and vertical displacements Δx and Δy between the pen down and succeeding points are compared with a predetermined minimum displacement, which is small enough to retain the essential details of sharp curves and cusps, if they should occur in that region. Our experiments showed that a sensible minimum displacement was 3 (see details of RATSAP-1 above). If neither of the displacements exceeds the parameter, the point is rejected. If the displacement exceeds the parameter, the point is conditionally retained for further evaluation with respect to the direction change criterion.

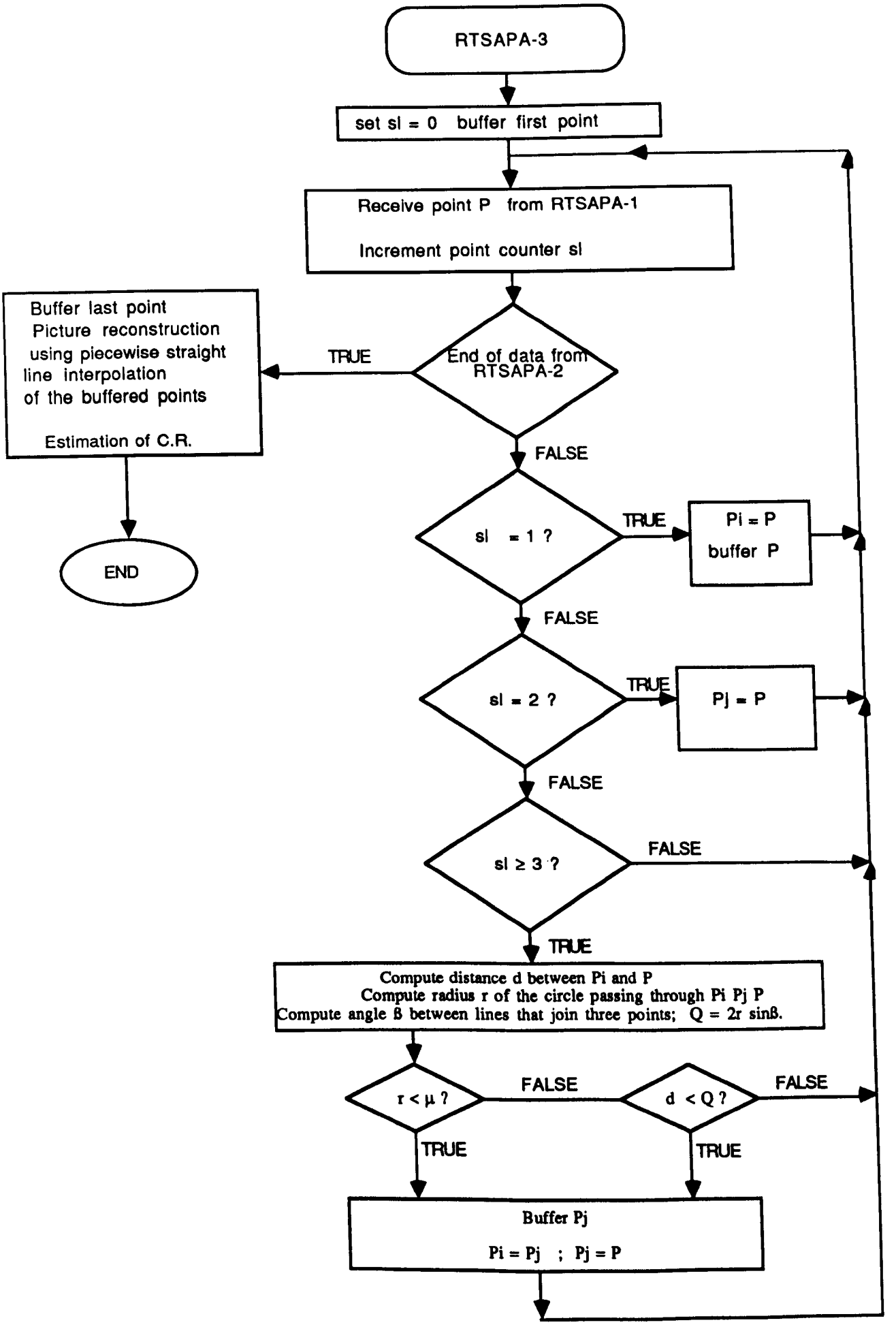
The direction change filter determines which of the points that have satisfied the minimum separation requirement are to be retained for defining significant curvatures or other changes of direction in the pen motion. The direction change filter may be implemented in either of the two ways described below:

In the first implementation, the angular directions of the displacements between the points that satisfy the minimum separation criteria are classified according to eight compass directions. When the direction of the displacement changes from one octant to another, the filter retains the point that preceded the change; otherwise, it rejects that point . In either event, the point just sampled is conditionally retained to determine whether it will satisfy the test for change of direction with respect to the next point.

In the second implementation of the direction change filter, a change in direction is determined by measuring the angle between the lines that joint three successive points and/or the radius of curvature of an arc passing through those points. First, the radius is calculated for the circle that passes through the point most recently retained by the filter and the next two points that satisfy the minimum separation criteria and have not been rejected by the previous filtering actions. If the radius thus measured is less than a predetermined value, the first of the aforesaid two points, that is the intermediate point in the series of three, will be retained. If this radius exceeds the predetermined value, then the angle between the two lines that joint the three points in succession must be examined. When the angle is less than a predetermined reference value, the intermediate point is rejected; otherwise it is retained by the filter. Experimentally, a suitable value of curvature threshold was 0.5, i.e a threshold radius of 2.

We have found that the above methods minimize the number of key points produced by the decimator, and that must be pronounced significant to provide an adequate image of the original hand produced material, when the significant points are connected by straight line segments. The first method tends to retain more points (perhaps unnecessarily) near octant boundaries, but it has the advantage of ensuring that even a very gradual change of pen direction into another octant will be detected and retained. The difference between the two methods is minor insofar as the quality of the handwriting display is concerned. However, in a commercial implementation, the choice would depend upon relative cost of implementation.

The flowchart of the computer program for RTSAPA-3 is shown in the following figure.


RTSAPA-3
set sl = 0 buffer first point
Receive point P from RTSAPA-1
Increment point counter sl
End of data from RTSAPA-2
TRUE
Buffer last point
Picture reconstruction
using piecewise straight
line interpolation
of the buffered points
Estimation of C.R.
END
FALSE
sl = 1 ?
TRUE
Pi = P
buffer P
FALSE
sl = 2 ?
TRUE
Pj = P
FALSE
sl ≥ 3 ?
FALSE
TRUE
Compute distance d between Pi and P
Compute radius r of the circle passing through Pi Pj P
Compute angle ß between lines that join three points; Q = 2r sinß.
r < μ ?
FALSE
d < Q ?
FALSE
TRUE
TRUE
Buffer Pj
Pi = Pj ; Pj = P

When the data representing a pen trajectory leaves the tablet, its journey is summarized in the figure shown on the following page. For reasons of increased data reduction, the above second method for direction change filter was used, so the following results were produced by it. The parameters required for the direction change filter are the estimation of the radius of curvature and the deviation angle between two successive vectors. Given three points P_{i-1} , P_i , P_{i+1}, it can be shown that (CRAMPING85), the radius is related to the area of the triangle by

$$\text{radius} = (|P_{i-1}P_i|\ |P_iP_{i+1}|\ |P_{i-1}P_{i+1}|) / (4\text{area}) \qquad (5.6)$$

$$\text{area} = |(y_{i+1} - y_{i-1})*(x_i - x_{i-1}) - (y_i - y_{i-1})*(x_{i+1} - x_{i-1})|$$

The angle ß between two successive vectors $P_{i-1}P_i$ and P_iP_{i+1} is related to the area of the triangle by

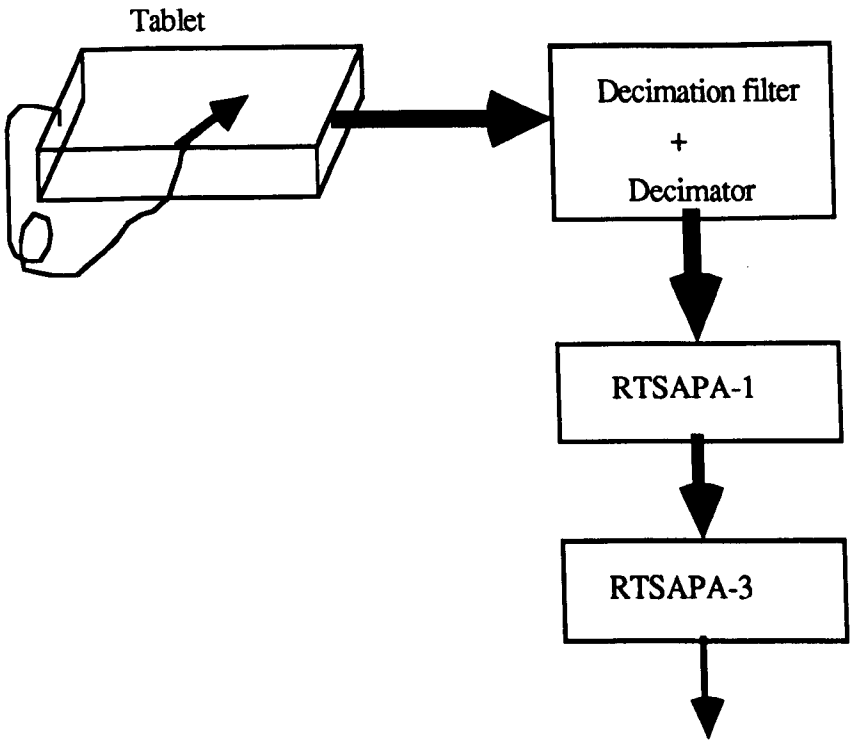
$$\text{area} = 2\ |P_{i-1}P_i|\ |P_iP_{i+1}| \sin ß \qquad (5.7)$$

So the first part of RTSAPA-3 takes only 2 subtractions; equation (5.6) indicates, that to avoid square roots, we can work in squares, so we have 12 multiplications, one division, and 3 additions, 7 substractions. As for equation (5.7), look up table techniques may be used to speed up the estimation of the angle ß. A suitable table of estimated sin ß can be generated, and we only have to calculate the expression

area / (2 $|P_{i-1}P_i|$ $|P_iP_{i+1}|$), which requires one multiplication, one division if squared up; because most of quantities involved have been

already calculated, when estimating the radius in (5.6).

So for each point from RTSAPA-1, the direction change filter may require 13 multiplications, two divisions, 3 additions and 7 subtractions.



The tests on data derived analytically are visually more successful than the results obtained for RTSAPA-2; this is confirmed by inspecting and comparing Figures 5.13 and 5.19; 5.15 and 5.20 for the analytical curves. Similar conclusions are drawn for the experimental data (Figures 5.17. and 5.21). These better results are obtained at the expense of a slightly lower data reduction rate.

5.2.3.1.RTSAPA-3 Evaluation based on accuracy criteria.

Observations similar to those for RTSAPA-1 and RTSAPA-2 have been made. A better visual accuracy was obtained for lower point elimination performance.

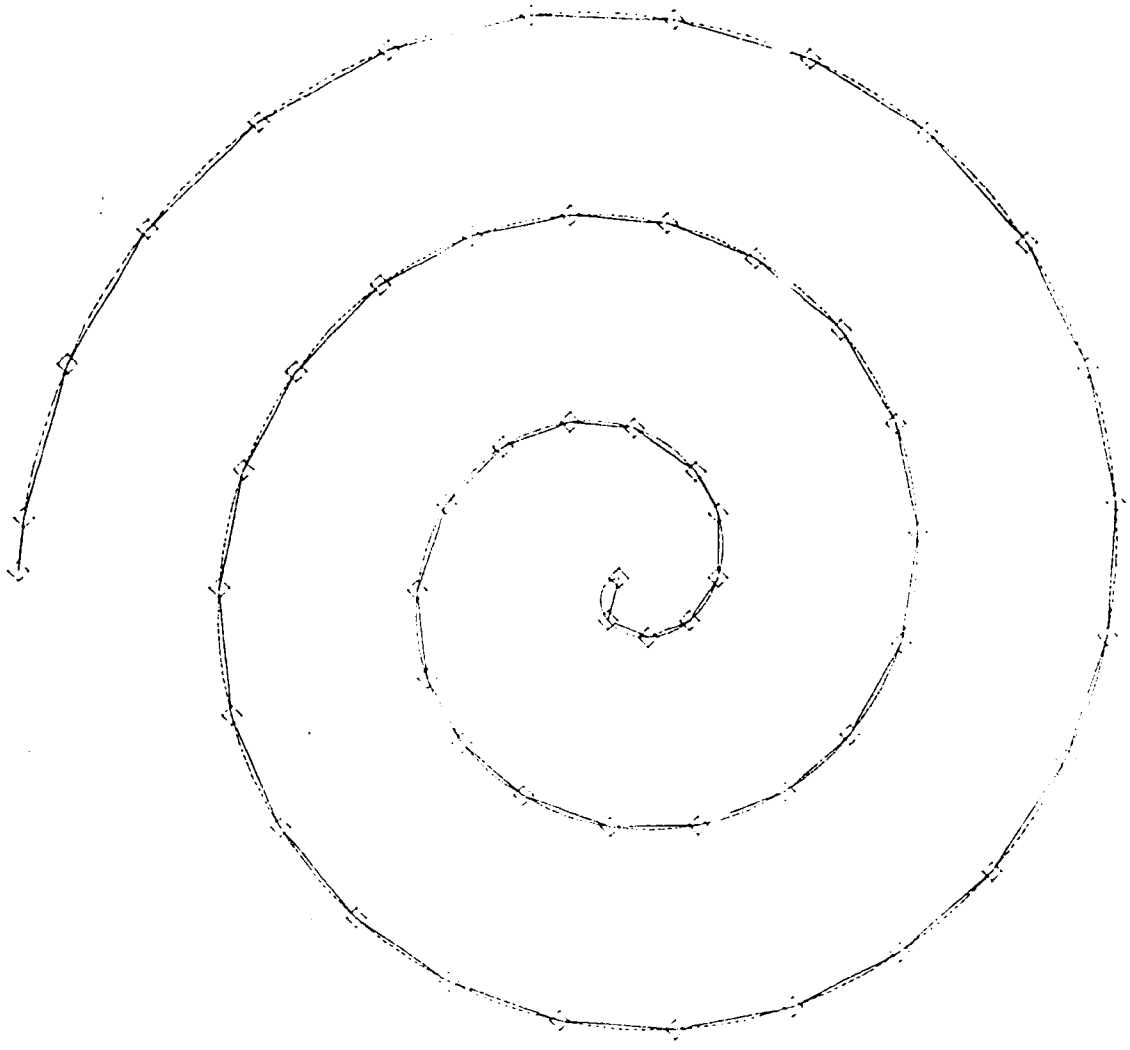


Fig.5.19
RTSAPA-2 approximation of Archimedes spiral. Maximum deviation from the piecewise straight line interpolator, is E = 1 unit of the output device (i.e Hewlet Packard plotter 7225)

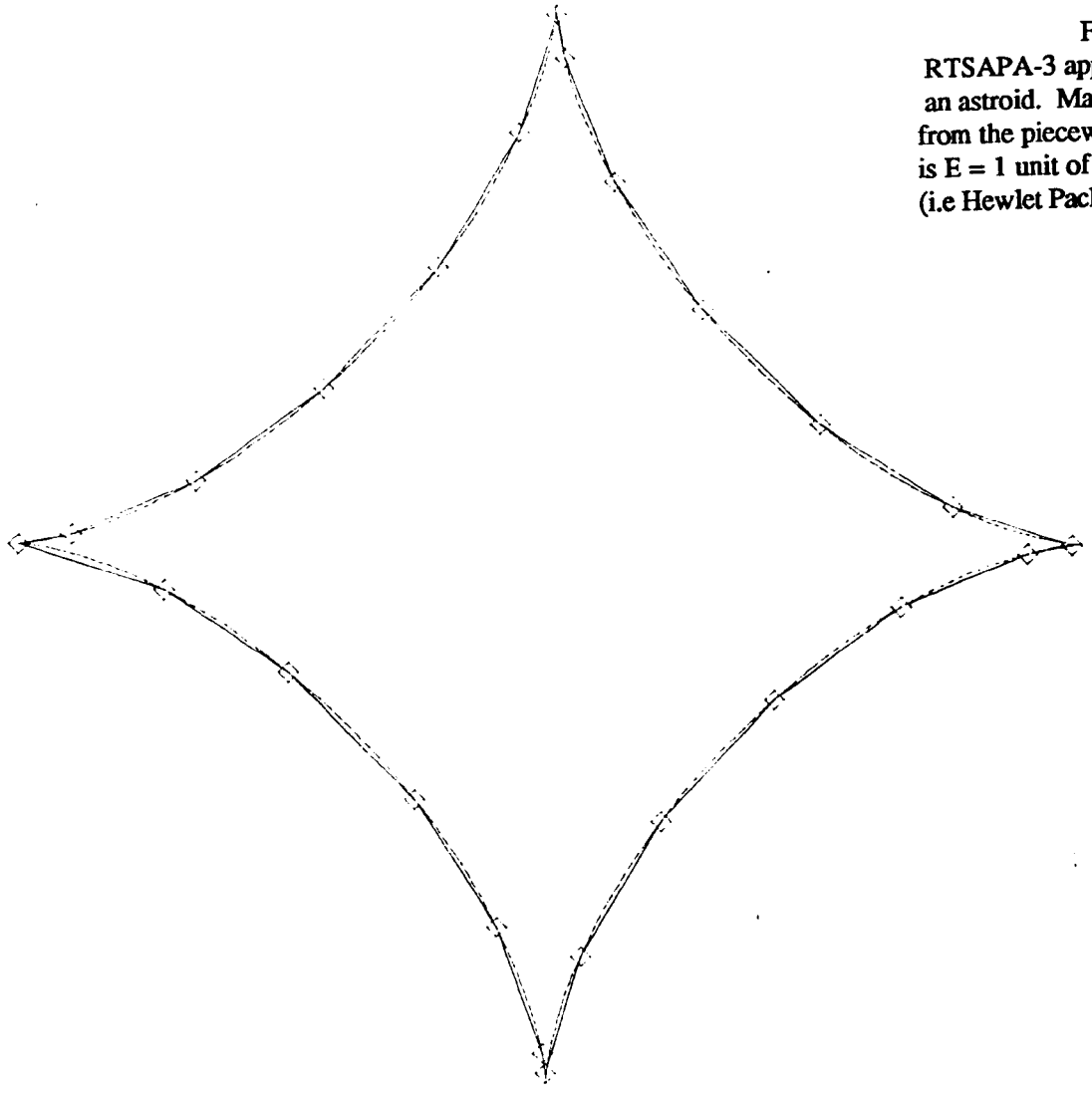


Fig.5.20
RTSAPA-3 approximation of
an astroid. Maximum deviation
from the piecewise straight line interpolator,
is E = 1 unit of the output device
(i.e Hewlet Packard plotter 7225)

In 1960 A. Renyi said:

"The fact that information can be measured is by now generally accepted

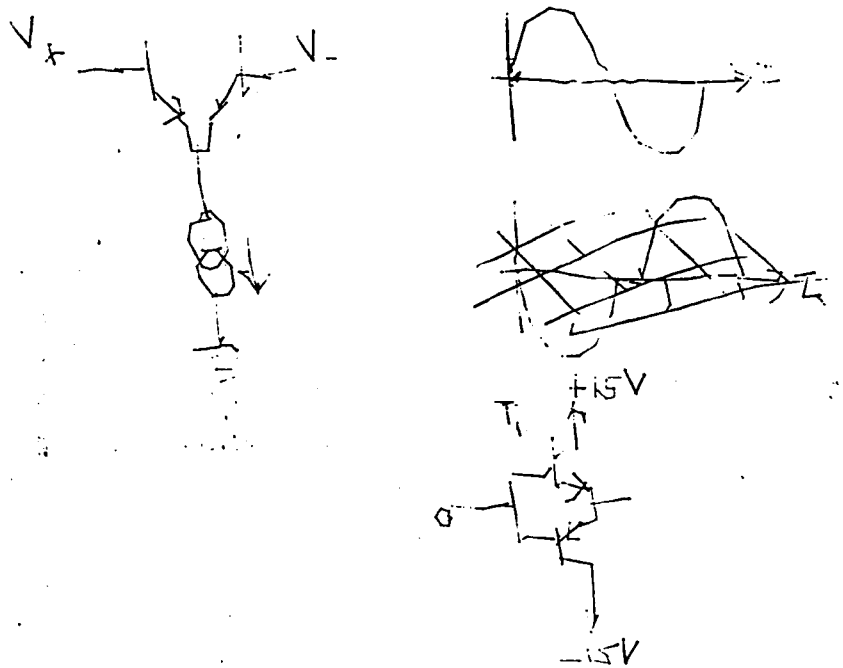


Fig.5.21
RTSAPA-3 approximation of handwriting and drawing. Maximum deviation from the piecewise straight line interpolator, is E = 1 unit of the output device (i.e Hewlet Packard plotter 7225)

5.2.3.2.RTSAPA-3 Evaluation based on efficiency criteria.

This algorithm requires 5 subtractions,two additions, and six multiplications for the processing of each incoming point. On the VAX/750 computer we have found that on average it takes about 1.5 seconds to process the picture displayed on Fig.5.1. which is made of 21736 points thus the points were processed at a rate of 14491 points per second.

5.2.2.3. RTSAPA-3 Evaluation based on compactness criteria.

The compactness performance of RTSAPA-3 is slightly higher than that of RTSAPA-2. In terms of compression ratio RTSAPA-3 is better than RTSAPA-2 by about 4 %. This can be checked by studying the relevant compression figures in Table 5.2 and Table 5.4. This is further discussed in the following section, dealing with the statistical properties of the signal represented by the successive relevant data point differences.

5.3. COMPARISON OF THE THREE ALGORITHMS.

The three criteria by which the algorithm can be evaluated and compared are:

1. Accuracy of the representation.
2. Efficiency of the representation.
3. The compactness of the representation.

The first criterion deals with the amount of error introduced by the process of choosing the best representative points of the pen trajectory. In so far as this criterion is concerned the three algorithms behave alike; the greater the accuracy tolerance, the greater the error during reconstruction. We have been concerned with the type of accuracy we call "visual accuracy".

The second criterion is designed for comparing the algorithm execution times. Our representation is intended for use in real time, so we are most concerned with the time of the algorithm responsible for choosing the points

which matter, along the pen trajectory.

But speed is the crucial question. We know that, after decimation the point acquisition takes place every 0.035 seconds. As long as the total time taken by the relevant point choosing process and the sequential reconstructing process is less than 0.035 s, we shall be happy to say that the algorithm may operate in real time.

Assuming a 32 bit microprocessor based processor, driven by a 16 MHz clock, for example the Motorola MC68020; if the calculations are done in software, we need 60 microseconds for each floating point multiplication, 70 microseconds for each addition. 70 microseconds for each subtraction, 125 microseconds for each division. These times are estimated on the basis of a 64 bits double precision floating point format. Using these computing time estimates, we can work out an estimate of the execution time for each algorithm. Table 5.3 shows a comparison of execution times.

From the table, we can see that the processing of a point may require 140 μs for RTSAPA-1, 975 μs for RATSAPA-2 and 1380 μs for RTSAPA-3. From the point of view of speed, RTSAPA-1 is 85% faster than RTSAPA-2 and 90% faster than RTSAPA-3; this is because RTSAPA-1 performs no multiplications or divisions.

The effective point acquisition period is 35000 μs. If economics dictate the software implementation option, and considering fast reconstructions techniques are available, eg DDA technique, Digital Differential analyzer, (FOLEY82), all three polygonal approximation methods may work in real time. Let us assume that a straight line interpolation is used for pen trajectory reconstruction, we can illustrate the DDA technique, i.e forward difference technique as follows:

A linear equation may be represented by

$$p(t) = ct + d \qquad (5.8)$$

Consider evaluating p(t) for n+1 equally spaced values of t. Thus we want to find $p_i = p(ih)$ where $h = 1/n$, $0 \leq i \leq n$. We can notice that the difference between two successive values of p(t) is constant; i.e $p_{i+1} = p_i + c*h$. So initially the quantity c*h is expressed once and for all, and the successive interpolating points are found by adding the constant c*h.

This technique will require 2 multiplications during the initial calculation of the constant c*h. After this, only two additions are used to generate an interpolating point. So n+1 points requires 2(n+1) additions and two multiplications carried out when the line is started off. As we shall see in a later chapter on higher order polynomial interpolation, the technique can be readily extended to nth order polynomial generation.

Considering the software implementation time estimates of RTSAPA-3 from Table 5.3; working in terms of μs (microseconds), real time processing may be possible if

1380 + 120 + 2*(n+1)*70 + graphic display processing time ≤ 35000

Here the graphic display processing time accounts for the processing time to address a screen buffer such as the MATROX board (SMOL81). Should the speed be the most important factor, a floating point coprocessor chip such as, Mc68881, would drastically speed the floating point arithmetic, the estimated figures can be found in the rightmost column of Table 5.3

The third criterion is concerned with the measure of data reduction achieved by the point choosing process. RTSAPA-2 is superior to RTSAPA-1 and

RTSAPA-3 in terms of data reduction while maintaining a reasonably recognisable picture.

In terms of visual acceptability, speed and compactness, RTSAPA-3 behaves much better than the other two algorithms, and is therefore the best one. A great many trials on various pictures has shown that RTSAPA-3 eliminates highly variable, non essential details, and preserves the significant features in both handwriting and drawing for angle ß <= 20 degrees. The next section deals with the statistical properties of the signal processed by RTSAPA-3.

Estimates of algorithms computing times relative to the processing of one data point

Algorithm	additions	subtractions	Multiplications	Divisions	Time (µs) Software implementation	Time (µs) Floating point chip coprocessor (Mc6881)
RTSAPA-1	0	2	0	0	140	38
RTSAPA-2	0	5	5	0	975	195
RTSAPA-3	0	5	13	2	1380	455

Table 5.3. Performance of algorithms computing times.

The above computing time estimates assume that the algorithms have been coded in "C language". We think that these computing times may be reduced if the algorithms are coded in " assembly language " of a dedicated processor.

5.4. Entropy rate estimation of the signal processed by RTSAPA-3

This section deals with the statistical structure of the signal approximated by RTSAPA-3. It is assumed that the whole static database represents a long signal. The reader is reminded that 56 data files which represent 13 tutorials produced by 13 different tutors were concatenated i.e. joined together to make the whole database. The whole database representing the electronic signal is processed as indicated in Fig.5.6. The statistical properties of the successive relevant data point differences are exploited in order to reduce the quantity of data required to convey a given amount of information. Results obtained by varying ATH are displayed on Table 5.4. For various settings of ATH, NSP, the total number of relevant data points, is listed in the second column. The compression ratio and the data reduction rate are respectively in columns 4 and 5.

The 'pen down state' average sampling rate (ie assuming continuous writing) listed in column 6, is defined by

$$\mathbf{avsr = NSP / PDT} \qquad \mathbf{(5.11)}$$

where PDT is the pendown time associated with the trajectories of the pen.

In the last column the effective average sampling rate is defined by

$$\mathbf{eavsr = NSP/ RECT} \qquad \mathbf{(5.12)}$$

where RECT is the recording time of the signal associated with the whole database. PDT and RECT were respectively 6419.38 seconds and 9878.80 seconds.

The visual perception in column 2 has been established on the basis of approximating various pictures by RTSAPA-3, and inspecting the results to see how visually satisfactory they were.

ATH (degrees)	Visual perception	NSP	C.R	D.R.R (%)	avsr (Hz)	eavsr (Hz)
0	good	126267	1.67	40	19.67	12.78
5	good	92139	2.30	56	14.35	9.33
10	good	79433	2.65	62	12.37	8.04
15	good	70789	2.98	66	11.03	7.17
20	good	64538	3.27	70	10.05	6.53
25	fair	59770	3.53	72	9.31	6.05
30	poor	54834	3.85	74	8.54	5.55

Table 5.4 Compactness performance of RTSAPA-3

From Table 5.4, we can see that, increasing the angle threshold leads to:

1 Higher compression ratio.

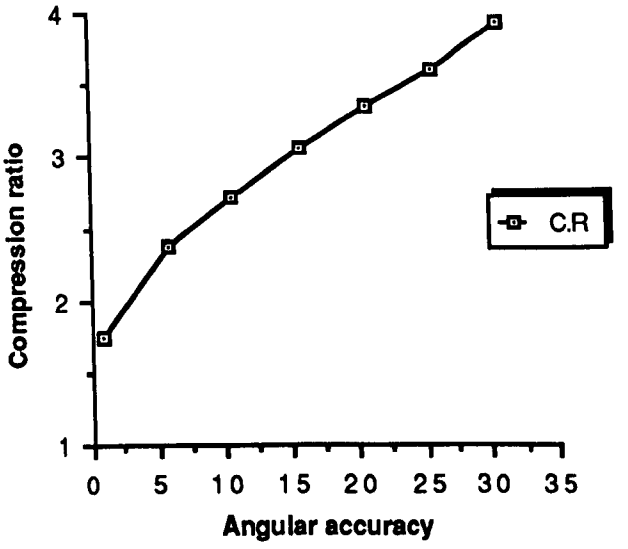
2 Higher data reduction rate.

3 Lower pen down state average sampling rate

4 Lower effective average sampling rate

5 Higher maximum error during picture reconstruction; thus the visual quality of the picture progressively deteriorates.

The graphical presentation of compression ratio against angular accuracy is shown the following picture.



The picture clearly, shows that the compression ratio grows with angular accuracy (ATH). But, straight line approximated freehand drawn material becomes unacceptable for angular accuracy greater than 20 degrees.

The compression ratio has a " meaningful " value, only if the picture recreated from the reduced data conveys the intended "communication".

Given that, the recreating process is a straight line interpolator, the compression performance is only useful for ATH $\leq$ 20 degrees.

The first order entropy estimates of the signal differences resulting from the RTSAPA-3 representation are presented in Table 5.5. for various settings of the angle threshold ATH. Using the strategy discussed in chapters 3 and 4, the entropies shown were computed from the corresponding histogram of Δx / Δy, for the chosen ATH.

The entropy rate estimate of the signal is calculated as in chapters 3 and 4, with the only difference that, within the context of this chapter , the probability distributions of the quantized difference signal are referred to the successive differences between relevant data points.

Using the formulae developed in chapter 3, the entropy rate estimates were calculated and displayed in Table 5.5.

We can see that the higher the angle threshold ATH, the lower the zero order entropy rate estimate. The same trend applies to the first order entropy rate estimate, calculated by taking into account the first order correlation between successive relevant data points.

Column two shows the average sampling rate associated with the pen being still or moving on the writing surface.

The first order entropy H_1 per $(\Delta x,\Delta y)$ expressed in bits is presented in column three. We can note that H_1 increases with ATH; but what matters in terms of transmission is the entropy rate which decreases as can be seen in column 4. The entropy decreases because the average sampling rate (avsr) decreases more rapidly than the entropy H_1 increases. This property agrees with SHAN48's transmission system capacity

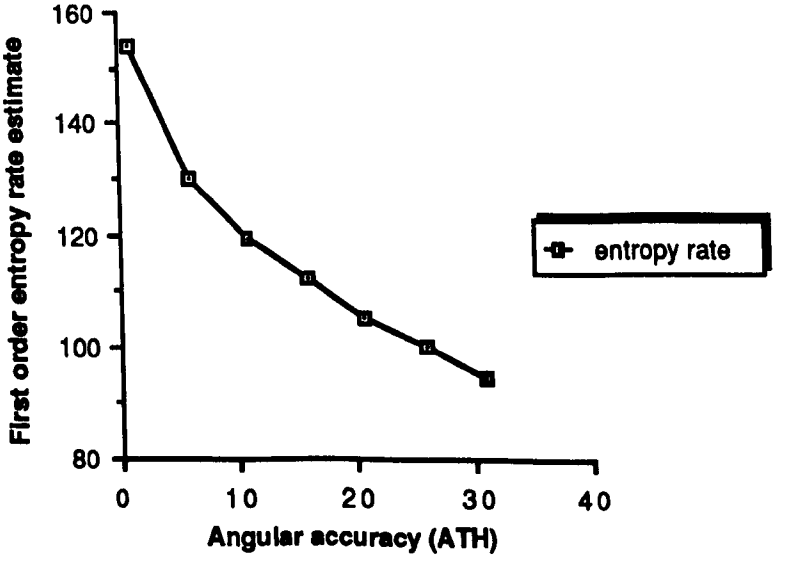
$$C = B \log_2 (1 + S/N)$$

where S/N is the signal power to noise ratio density, and B is the bandwidth. (in our case $B \approx avsr / 2$) This formula suggests that if the bandwidth decreases more rapidly than $\log_2 (1 + S/N)$, the bit rate C should decrease.

ATH	avsr (Hz)	H_1 (bits)	Theoretical entropy rate (bits/s)	practical limit entropy rate (bits/s)
0	19.67	6.68	131	152
5	14.35	8.04	115	128
10	12.37	8.62	107	117
15	11.03	9.15	101	110
20	10.05	9.45	95	103
25	9.31	9.78	91	98
30	8.54	10.08	86	92

Table 5.5 First entropies estimates for various settings of angular accuracy.

The following picture is a graphical presentation of the first order entropy rate estimate against the angular accuracy ATH. The entropy rate falls progressively for increasing values of ATH. The falling entropy rate does not necessary imply that the quality of the reconstructed material, is good. This is understandable because the entropy concerns not the question " What sort of information ? " but rather "How much information ?". Our experiments indicate that the entropy is meaningful for angular accuracy ATH ≤ 20 degrees, because the regenerated material conveys the originally intended "communication".



A more detailed analysis was carried out for ATH = 20 degrees. This angle threshold was taken as a baseline for further analysis because, it provides optimum results in terms of visual perception, and compression ratio (ie compactness).

The relative frequency of occurrences of classes of relevant data point differences in the x and y directions are the estimated probabilities which are used to calculate the signal entropy rate estimates.

Assuming x-y correlation, the first-order statistical analysis of the signal samples successive differences, yields 3600 classes of (Δx, Δy). In principle, a second order statistical analysis may involve 3600^2 conditional and joint probabilities. This represents a serious computational problem. Should one contemplate higher order statistical analysis, the computational problem would become even more disturbing, because it grows exponentially. Hence, as in the previous chapters, no x-y correlation is assumed. The entropy rate of the signal is estimate from separate nth order statistical analyses of Δx and Δy, for $0 \leq n \leq 8$. The following table gives the first order probabilities of $\Delta x/\Delta y$.

Δx/Δy	p(Δx)	p(Δy)
-10	0.00764	0.00823
-9	0.00965	0.00878
-8	0.01165	0.12873
-7	0.01265	0.01638
-6	0.01685	0.01835
-5	0.02085	0.02215
-4	0.02967	0.02937
-3	0.04365	0.04286
-2	0.06945	0.07049
-1	0.10965	0.11887
0	0.34956	0.35092
1	0.07981	0.08715
2	0.04943	0.05244
3	0.03477	0.03503
4	0.02312	0.02256
5	0.01864	0.01653
6	0.01386	0.01158
7	0.01185	0.01054
8	0.00997	0.00965
9	0.00806	0.00867
10	0.00576	0.00613

Table 5.6 Probability distributions of Δx/Δy generated from RTSAPA-3 output

The range of Δx was from -40 to +52, and that of Δy from -62 to 66; the lowest and highest probabilities of occurrence were 0.00001 and 0.34956 for Δx, 0.00001 and 0.35092 for Δy. Five significant figures after the decimal point were reasonable, because the total number of samples 64538, was between 10^4 and 10^5. Table 5.6 shows only the relative frequencies of $-10 \leq \Delta x \leq 10$ and $-10 \leq \Delta y \leq 10$ because 2 % of Δx out of the given range, have relative frequencies less than 0.005 and 2 % of Δy out of the given range, have relative frequencies less than 0.005. Because those Δ's are rare, their contributions to the total entropy are not significant.

As explained in chapter 3, an appropriate estimate of the signal entropy must be found by taking into consideration the higher order correlation between successive vectors (ie first differences (Δx,Δy) relevant data points produced by RTSAPA-3 with ATH = 20. For the nth order correlation (n = 0 to 8), the entropies and entropy rates are presented in Table 5.7.

Table 5.7 assumes an average sampling rate of 10.05 Hz (see Table 5.5).

n	H_n (bits)	Theoretical bit rates (b/s)	Practical limiting bit rates (b/s)
0	13.44	135	140
1	10.09	100	113
2	8.39	84	93
3	7.43	75	87
4	6.60	66	82
5	5.84	59	78
6	5.20	52	75
7	4.60	46	73
8	3.85	39	70

Table 5.7 Theoretical bit rates and practical limiting bit rates

Table 5.7 reveals that :

1 The order, n, of the correlation between successive vectors is listed in column 1.

2 For various values of the order of correlation, the entropy per element of the signal is in column 2; this is the theoretical average number of bits which may be used to represent (Δx,Δy) and is found on the basis of the knowledge of a sequence of previous vectors. The details of calculations were discussed in chapter 3. As expected, the table indicates that the higher the correlation, the lower the entropy per signal element.

3. The entropy rate estimate of the signal decreases with higher order correlation, this is highlighted in column 3. The ultimate upper bound entropy rate estimate of the signal is 39 bits per second which is calculated when 8 previous vectors (i.e 9 previous relevant points) are known.

4. As in previous chapters, we endeavour to calculate the practical limiting entropy rates from the

theoretical results of Table 5.7. Pen trace length statistics, of course new lengths of significant points selected by RTSAP-3, were used to estimate the bit rates that might be used in practice. To remind the reader, if the order of correlation, n is 8, we have 18 bits for the first point of a pen trace, H_1 bits for the second point, the third H_2 bits and so on, proper averaging, as explained in chapter 3 and 4, was carried out over segments, and the results shown on the right hand column of Table 5.7 were found. It is clear that the target bit rate of at most 200 bits/s for graphical transmission over conventional telephone lines, may be satisfactorily met, by considering only the first order correlation.

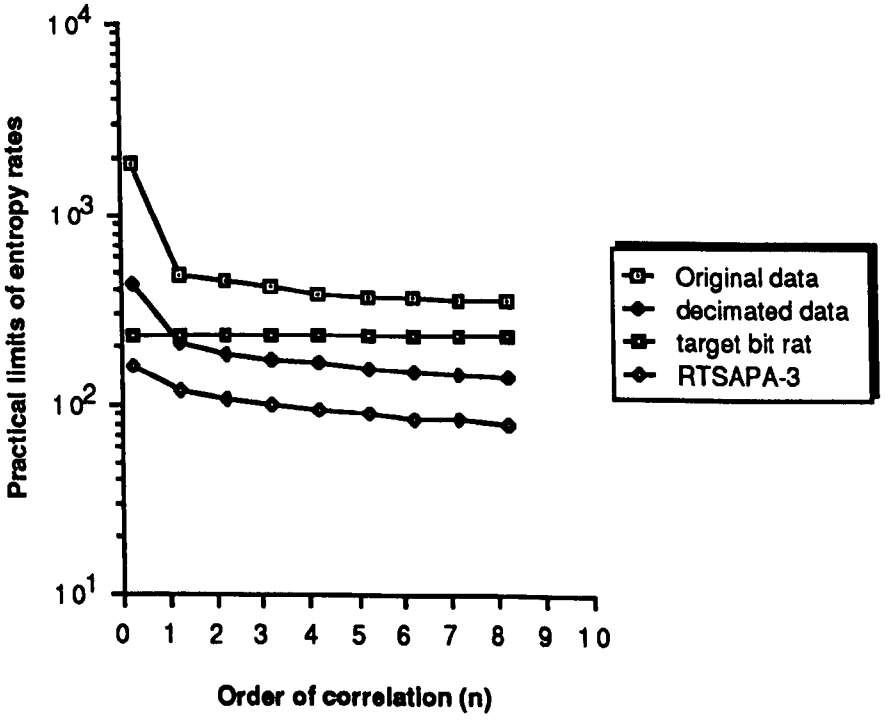
5.4 Comparison of entropy rates of the signals, estimated from original and approximated.

Fig.5.22 is a graphical presentation of signal entropy rate estimates, associated with original data and reduced data output by, a 7 to 1 decimating process and RTSAPA-3. Four entropy rate curves are brought together for comparative purposes :

1. The flat curve is the target entropy rate (i.e 200 bits / s).
2. The curve associated with original is above the target entropy rate curve. The lowest entropy rate estimate of the signal, calculated from the original data is above the target bit rate 200 bits/s by 57 %.
3. The entropy rate curve associated with the decimated signal drops below 200 bits / s from first order correlation. The lowest entropy estimate occurs at 8th order correlation and is below the target bit rate 200 bits/s by 38 %.
4. From zero order correlation, the entropy rate curve associated with RTSAPA-3 is progressively below the target entropy rate curve.
   The lowest entropy rate estimate is below 200 bits/s by 65 %.
   The highest entropy rate estimate associated with RTSAPA-3 occurs at zero order correlation and is 30 % below 200 bits/s .

From the above discussions, the most effective entropy rate curve is associated with RTSAPA-3 for two main reasons :

1. The quality of the approximated hand generated material is visually good enough.
2. A simple coding based on first order frequency distributions of data approximated by RTSAPA-3, should lead to a graphic signal bit rate which is small enough, to enable the simultaneous transmission of handwriting and speech signals over a single telephone circuit.



5.6 CONCLUSIONS

In this chapter we have constructed two new algorithms for economizing on the number of line segments needed to approximate the freehand generated curves represented by n points (x,y). They are sequential algorithms, their computing time is of the order of O(N), where N is the number of points of a picture.

RTSAPA-2 and RTSAPA-3 yield excellent results in economizing on the number of line segments required to represent data, and in so doing they are a good tool in the bandwidth reduction of pictures which result from human freehand generated curves. The development of RTSAPA-3 and RTSAPA-2 results from RTSAPA-1

The quasi-optimum calculations of the estimate of the entropy rate of the quantized handwriting have been calculated by appropriately using the nth order (n = 1..8) probability distributions of the successive differences of relevant data points produced by RTSAPA-3 (ATH = 20 degrees).

The illustrative figures have shown how variations in the accuracy tolerance parameter can affect the degree of bandwidth compression as well as the visual quality of graphic material which represents handwriting and drawing.

It must be noted that this chapter pertains to the approximation of data by means of straight lines. It is thought that higher order polynomial

(e. g cubic splines, Hermite polynomials) may generate a smoother reconstruction of the segmented handwritten or drawn material. This is dealt with in a later chapter.

6. SELECTION OF SIGNIFICANT PEN POSITIONS, BY METHODS WHICH MAY NOT WORK IN REAL TIME.

As pointed out in chapter 1, electronic tutorials may be stored on a recording medium and transmitted "off - line". This kind of requirement is characteristic of distance teaching, where hand generated material may be stored and replayed several times, to cater for the time tables of various students. Indeed, in any situation, where immediate transmission is not required, material may be prerecorded and transmitted at a later date, and we shall refer to this as "off -line" processing of hand generated material.

In "off -line" processing conditions, the data describing various trajectories of the pen are available and stored on a recording medium; thus for each pen trace, all the pen locations are known simultaneously. Because, all information about a trace is known at the same time, the selection of relevant points may be carried out optimally. An optimal or quasi optimal algorithm for selecting significant points from a set of points describing a trace, might require a great deal of processing, consequently the computing time might be too high for real time use. In this chapter, we describe algorithms for selecting relevant points, which may not work in real time. Though there might be many practical cases where the delay required for processing of each line segment would be acceptable.

6.1 Introduction

As in the previous two chapters, this chapter is concerned with the problem of selecting relevant points from a set of points which make up a pen trace. The digitized trajectory of the pen is represented by a two dimensional array of points, and it is desired to extract from the given array a set of

points from which the path of the pen can be reconstructed by interpolation using either a first order polynomial, i.e polygonal approximation or a third order polynomial (e.g cubic spline). So we are concerned with three problems:

1. How to partition each pen trace into segments.
2. How to partition each segment into pieces.
3. What representation to use for each piece.

We consider that the location of breakpoints (i.e significant points) is the most important information rather than the description of the curve in between. Having said that, in this chapter, piece wise fitting of lines is used. Higher order polynomial fitting will be considered in the next chapter.

Let assume that a pen trace is partitioned into segments. Assuming that N ordered points make a segment; and given an error E, the partitioning of each segment into pieces consists of finding the least number of polygonal sides, for which the error is less than E.

The various stages of our optimal approximation are summarized in Figures 6.1.a, 6.1.b, 6.1.c, which depict respectively an original pen trace (6.1.a), two segments resulting from corner detection (6.1.b), and each segment is subsequently subdivided into smaller segments (i.e pieces) which may be approximated by lines (6.1.c).

To recapitulate, it is thought that a very efficient compression of hand generated material should require initial partitioning of a pen trace into segments. The definition of a pen trace (or trajectory of the pen) was given in chapter 2. Each segment is further partitioned into smaller segments which are then represented by suitable geometric primitives; e.g straight lines.

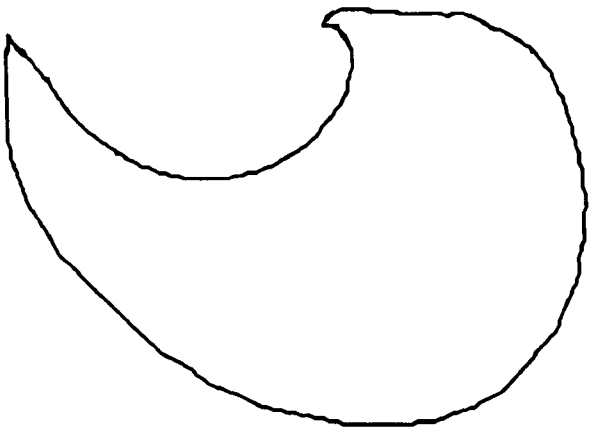


Fig.6.1.a. A pen trace.

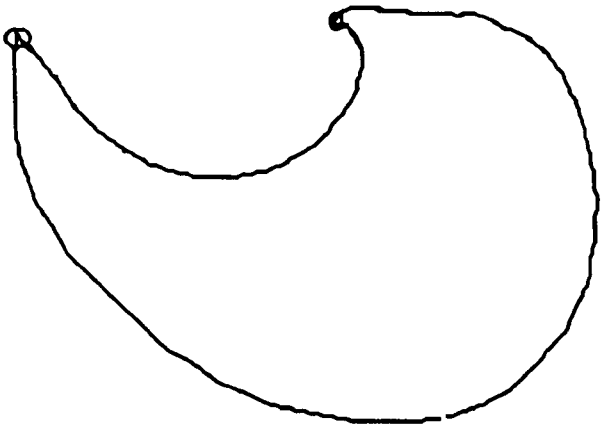


Fig.6.1.b. Two corners are found, highlighted by two small circles; thus the pen trace is partitioned into two segments.

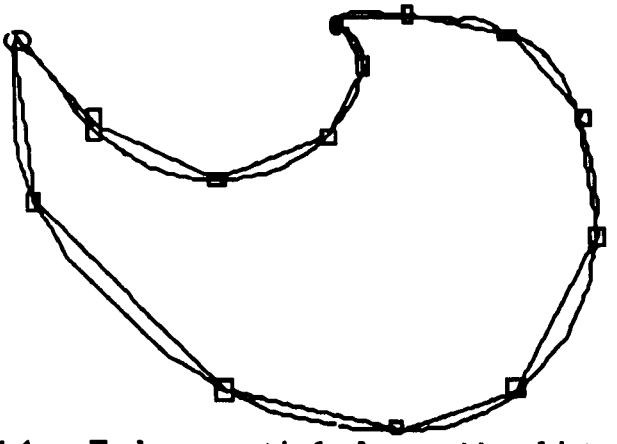


Fig.6.1.c. Each segment is further partitioned into pieces which may be approximated by straight lines. Each piece is delimited by two squares.

The remainder of this chapter is structured as follows:

1. Pen trajectory partitioning methods; for each pen trajectory this section considers the creation of what we may call high level segments. Each high level segment is further segmented by quasi-optimum significant point selection methods discussed in the next section.
2. Discussion of a suitable point selection method.
3. Initial consideration of higher order polynomial (e.g cubic) descriptions of pen trajectories.

6.2 High level partitioning

We consider the problem of defining the critical points for pen traces with various degrees of curvature. The methods described in this section attempt to detect the high level partitioning points which should belong to the following group :

1. End points of curves produced by the pen.
2. Points where the curvature changes sign.
3. Points where direction changes abruptly (corners).

As pointed out in the introduction, partitioning a pen trace calls for the detection of the corners from the digital representation of the trace. A corner is a significant angle. Finding a digital angle on a digital curve is not a trivial problem (ROSEN76), because a digital curve is polygonal, and one must distinguish between angles that are due to the discrete nature of the digitizing process and angles that represent significant changes in the direction of the curve.

In this section, we present two simple techniques for detecting corners on a digital curve.

A number of methods can be found in the literature (LANG82, SHAH84), but ours are either the refinement of existing work, or thought out in the

course of our work. Results obtained using these techniques are compared with each other and with subjective corner detection judgments.

6.2.1 Partitioning technique 1: k-curvature

This method was introduced by ROSEN73 and appeared to be the backbone of work published in recent years (FREEMAN78, SHAH84, KITCH83). ROSEN73 's work assumed that a digital curve is defined by a chain code (FREEMAN74); this is essentially a sequence of vectors

$V_0, V_1, ...V_{n-1}$, where V_i takes its components from the set :

{(1,0) , (1,1), (0,1), (-1,1), (-1, 0), (-1, -1), (0,-1), (1,-1) }

In our application, unless the motion of the pen is extremely slow on the writing surface, the successive vectors which describe the displacement of the pen, may have components out of the above range (i.e -1 to 1), thus our formulation of k-curvature was slightly different from the work described in ROSEN73. When starting off, the first and last point of a trace are taken as partitioning points. To find further partitioning points in a pen trace, we first calculate for each point, the change in angle of the two line segments joining the point to its neighbouring points. Thus, we let ϕ_i be

$$\phi_i = \arccos \left((V_i \cdot V_{i+1}) / (|V_i| \, |V_{i+1}|) \right) \qquad (6.1)$$

where $V_i = P_i - P_{i-1}$; $V_{i+1} = P_{i+1} - P_i$

This calculation is illustrated in Fig.6.2. If P_i and P_{i+1} are less than k units apart, then the next point P_{i+2} is used instead; this means that the magnitudes of vectors Vi and V_{i+1} must be at least k units. This prevents local noise from entering into the ϕ measure. The parameter k is dependent

on the accuracy and resolution of the digitizer and on the accuracy of the human operator. Increasing k smooths more noise; decreasing it can allow small perturbations to become more significant.

ROSEN73 and FREEMAN78 call the above ø measure, incremental curvature and use it for describing two-dimensional shapes.

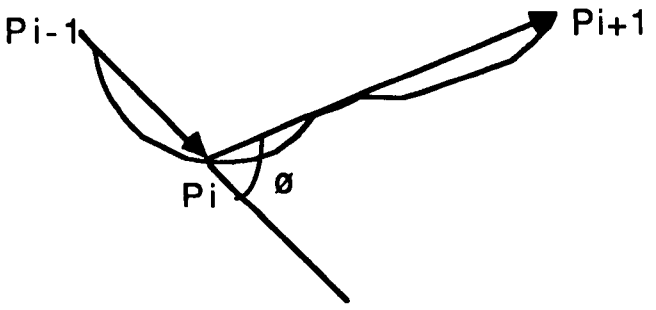


Fig.6.2 Deviation angle ø between two successive vectors.

Figures 6.3, 6.4 and 6.5 depict three test traces.

The incremental curvatures for Figures 6.3 and 6.4 were estimated by using equation (6.1) with k = 5 units (i.e 2.5 cm of Bit Pad).

The absolute values of the corresponding incremental curvatures are shown respectively in Figures 6.3.ø and 6.4.ø.

Fig.6.3.ø shows that the maximum deviation angle was less than 15 degrees, this was not significant enough to warrant the selection of any extra partitioning point in addition to a single point which represents the departure and arrival of the pen tip. As to Fig.6.4.ø, the graph indicates, the angle deviation may vary from to 0^{o} to 120^{o}. A careful study of this graph shows that angles above 80^{o} corresponds to regions of potential corners; now the question of isolating the real corner remains.

If many contiguous corners were detected in a section of a pen trace,

a straight line was fitted to them. The point associated with the highest deviation from the fitted straight line was retained as the true corner, and therefore was retained as a partitioning point.

Fig. 6.3
Test trace generated from the Bit Pad

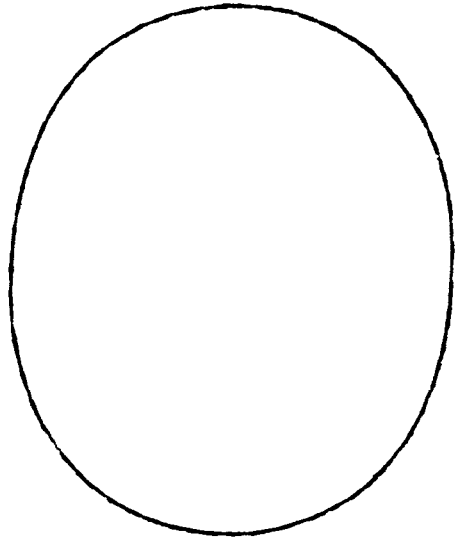


Fig. 6.4
Test trace generated from the Bit Pad

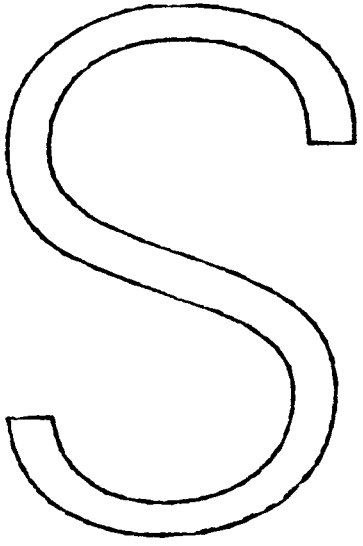
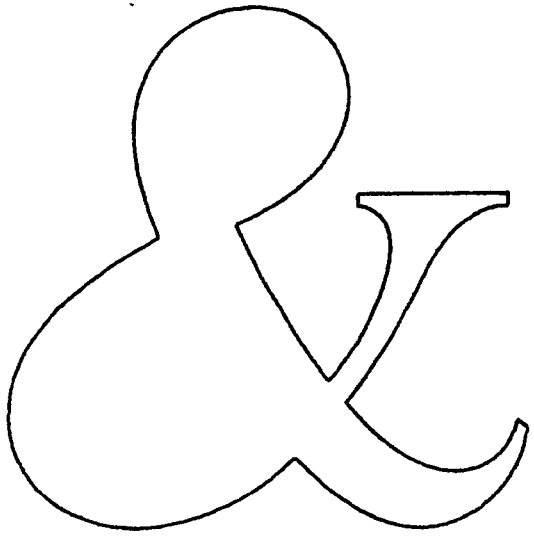


Fig. 6.5
Test trace generated from the Bit Pad



Applying this strategy led to 4 partitioning points for Fig.6.4.

This is shown in Fig.6.4.b, where the corners are depicted in red star symbols. The incremental curvature reflects changes in the directional displacement of the pen. Note how the partitioning points appear as spikes in incremental curvature curves (Fig.6.4.ø). This is because the partioning points are points where the pen changes direction very rapidly. The angle-oriented incremental curvature is used instead of a slope criterion because slope is unbounded from $-\infty$ to $+\infty$ and because changes in slope at a potential partioning points (e.g cusp) are dependent on the orientation of the cusp. The incremental curvature is bounded by -180^{o} to $+180^{o}$ and is independent of cusp orientation.

In searching for the partioning points of a pen trace, we search the incremental curvature (FREEMAN78) for points such that $|\phi_i| \geq \phi_{th}$. The variable ϕ_{th} is a tolerance measure.

After many trials, we found $\phi_{th} = 80$ degrees, detected all points that have a change in direction of greater or equal to 80^{o}.

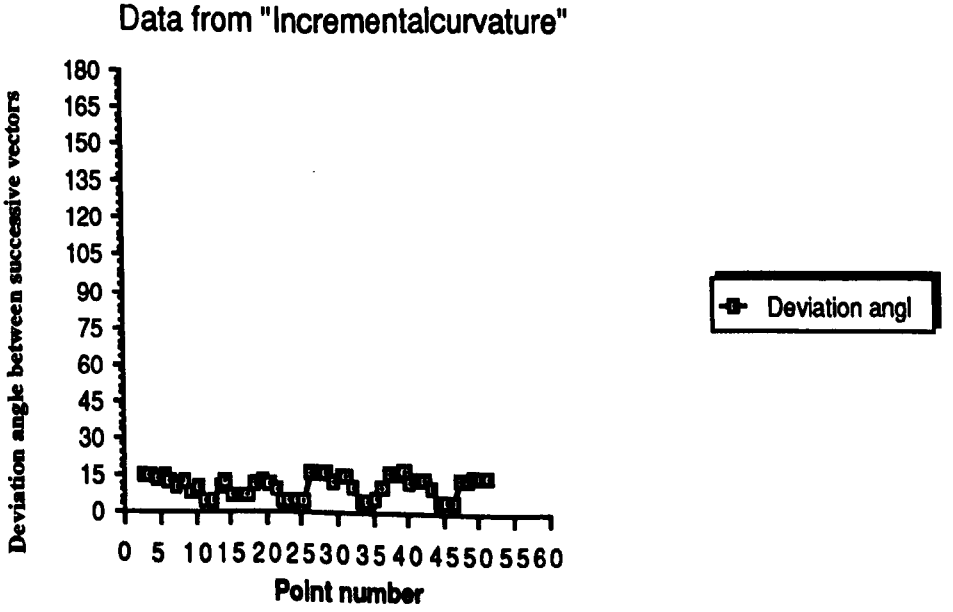


Fig.6.3.ø. Variations of deviation angle between successive vectors

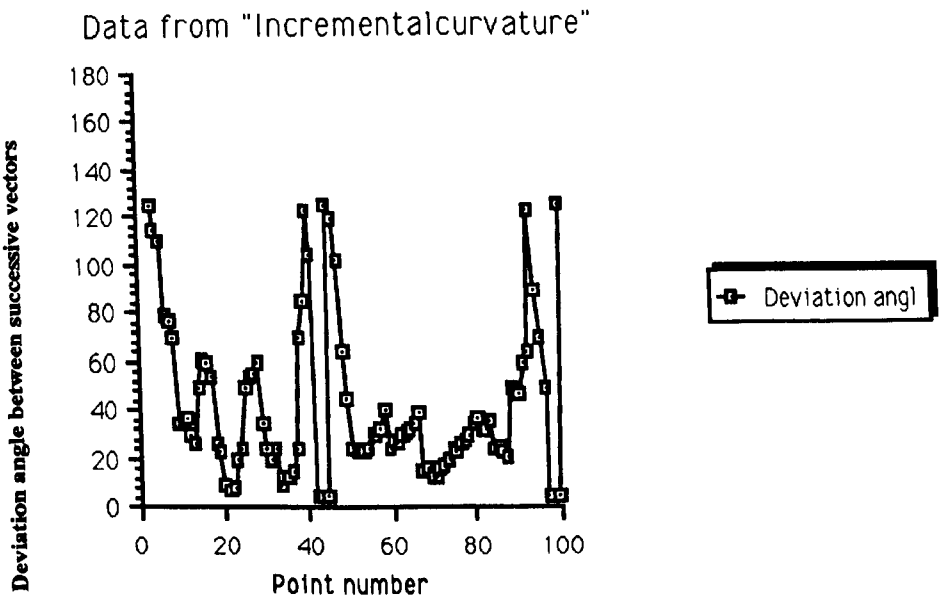


Fig.6.4.ø.

The variations of deviation angle between successive vectors for Fig.6.4 are shown in Fig.6.4.ø.

Our experiments have shown that, if we plot the incremental curvature, at an ideal corner, its shape looks gaussian.

Pratical results:

The pratical results for the k-curvature method speak for themselves; clearly a good performance depends upon the choices of k and the threshold angle ϕ_{th}.

A human eye would not detect a corner on the trace depicted in Fig.6.3; to humans, this trace is smooth and a suitable choice of k should lead to angle deviations near to zero, but if we look at the incremental curvature, we can see that for k = 5, the maximum deviation angle is 15^{o}. So any threshold angle less or equal to 15 degrees should lead to the detection of corners, which are obviously erroneous. Many trials have shown the true corners are detected when threshold angle is set to 80^{o}; thus there is no true corner for Fig.6.3; however an obvious high level partioning point would be the

starting point (Fig.6.3.a).

Fig.6.4.a shows the corners for Fig.6.4; here the red stars are used to indicate potential coner regions. We can clearly see that we get more than one corner, where a human should expect a single real corner, to deal with this problem, a straight line is fitted to each group of close corners, and the furthest point from the straight line is retained as a corner. This is illustrated in Fig.6.4.b which shows four expected corners (red stars), and an extra high level partitioning point which is the starting point of the trace.

Similar results are shown Fig.6.5.a and Fig.6.5.b for the original trace depicted in Fig.6.5.

It is noted that, once a true corner is selected, any point used for corner detection must be distant from any corner that has already selected by at least k units (i.e 0.5k mm of the Bit Pad SMOL81).

For the algorithm coded in "C language", the average computation time for this method was 7 seconds for the traces shown. This time may be shorter if the algorithm is coded in "assembly language" of a dedicated processor. However this speculation was not experimentally checked.

6.2.2 Partitioning technique 2 based on a fitted curve.

In the above partitioning technique 1, the incremental curvature at a given position is estimated by taking k points on either sides of that point. The k-slope can be estimated; and as the curvature is defined as the rate of change of the slope, the k-curvature can then be obtained by taking the differences between k-slopes on either side.

The reason for taking k points is to reduce the effects of quantization on

Fig.6.3.a
Segmenting point is the starting point.
No corners are found.

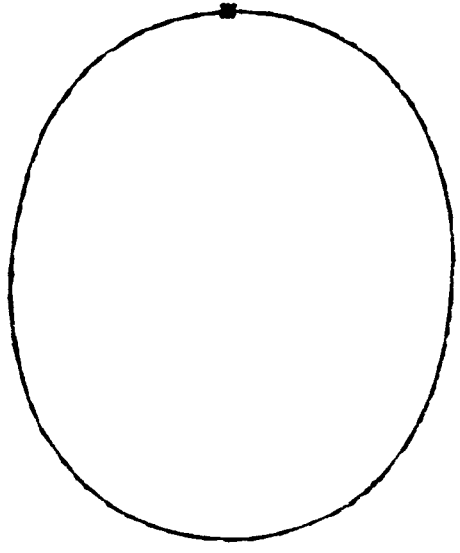


Fig.6.4.a
Higher level segmentation of test trace by k-curvature method. Too many corners are detected in the same region

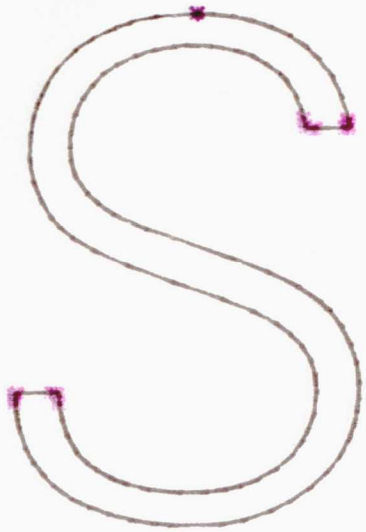


Fig.6.4.b
Higher level segmentation of test trace (Fig.6.4) through the detection of corners, which are labelled in red star symbols.

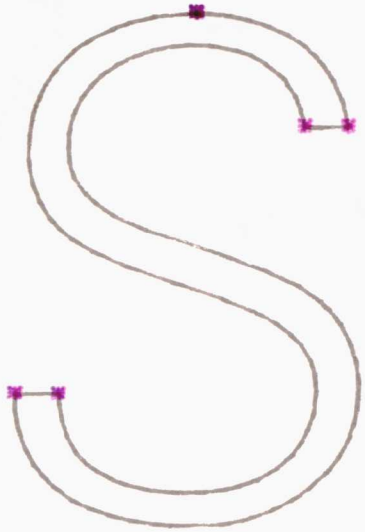


Fig.6.5.a
Too many corners are found in the
same vicinity. (k-curvature method)

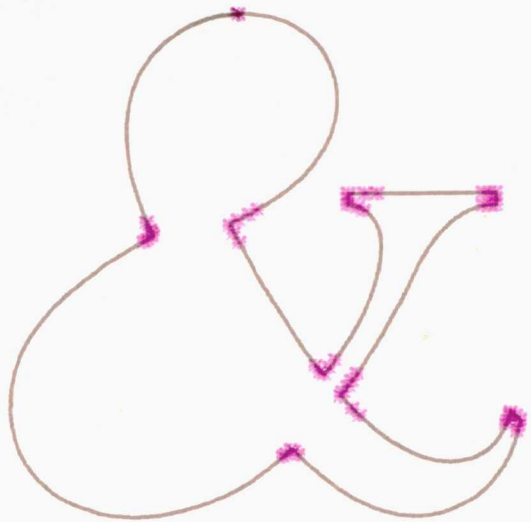
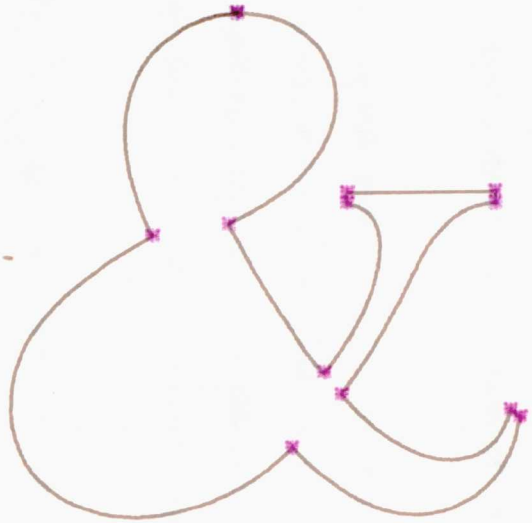


Fig.6.5.b
After filtering true corners are preserved
(k-curvature method)



slope or curvature, so k indicates the amount of smoothing. This is rather a crude smoothing technique. In order to use the usual analytical definition of the curvature, we need an analytical representation of the digital points; thus in this section, the partitioning points are extracted from the digital positions of the pen described by a cubic B-spline. A cubic polynomial description of the pen trajectory provides us with a continuous curve, and then, we can use the well known curvature expression, which is defined as the changing ratio of the tangent along the arc.

Curvature can be defined analytically as follows :

Consider a curve y = f(x), the definition of the curvature is

$$(d^2y / dx^2) / (1 + (dy/dx)^2)^{3/2} \qquad (6.2.a)$$

As some curves turn back onto themselves, to get around infinite slope, parametric representations seem rather appropriate.

Let the trajectory of the pen be described in parametric forms by $p(t) = (x(t), y(t))$. The first order derivatives at time t are

dy/dt, dx/dt; thus $dy/dx = (dy/dt) / (dx/dt)$ (6.2.b).

For the second order derivative d^2y/dx^2, we apply the derivative formula for a rational function u/v; i.e $(u' v - v' u) / v^2$ to (6.2.b). So

$$d^2y / dx^2 = (d^2y /dt^2 \, dx/dt - d^2x/dt^2 \, dy/dt) / (dx/dt)^2 \quad (6.2.c)$$

Substituting (6.2.b) and (6.2.c) into (6.2.a) and little algebraic manipulations lead to the curvature at time t

$$C(t) = \left((dx/dt)(d^2y/dt^2) - (dy/dt)(d^2x/dt^2) \right) / \left(((dx/dt)^2 + (dy/dt)^2)^{3/2} \right) \quad (6.3)$$

Assuming that x(t) and y(t) are cubic polynomials

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \quad (6.4)$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

Assuming that the time parameter t is normalized to vary between 0 and 1, considering equations (6.3) and (6.4); appropriate derivations of the derivatives lead to the curvature

$$C(0) = 2 (c_x b_y - c_y b_x) / (c_x^2 + c_y^2)^{3/2} \text{ at } t = 0 \quad (6.5)$$

To estimate the curvature at each pen position, the trajectory of the pen is smoothed using B-splines. From FOLEY82, the parametric cubic B-spline is defined by

x(t) = TMPsx

y(t) = TMPsy

where M is a 4 by 4 matrix, T is a 1 by 4 matrix,

Ps a 4 by 1 matrix of point coordinates.

$T = [t^3 \; t^2 \; t \; 1]$, and the other matrices are expressed below

$$M = \begin{bmatrix} -1/6 & 3/6 & -3/6 & 1/6 \\ 3/6 & -1 & 3/6 & 0 \\ -3/6 & 0 & 3/6 & 0 \\ 1/6 & 4/6 & 1/6 & 0 \end{bmatrix}$$

$$x(t) = T \begin{bmatrix} -1/6 & 3/6 & -3/6 & 1/6 \\ 3/6 & -1 & 3/6 & 0 \\ -3/6 & 0 & 3/6 & 0 \\ 1/6 & 4/6 & 1/6 & 0 \end{bmatrix} \begin{bmatrix} pxi-1 \\ pxi \\ pxi+1 \\ pxi+2 \end{bmatrix}$$

$$y(t) = T \begin{bmatrix} -1/6 & 3/6 & -3/6 & 1/6 \\ 3/6 & -1 & 3/6 & 0 \\ -3/6 & 0 & 3/6 & 0 \\ 1/6 & 4/6 & 1/6 & 0 \end{bmatrix} \begin{bmatrix} pyi-1 \\ pyi \\ pyi+1 \\ pyi+2 \end{bmatrix}$$

Computing TM first, little algebraic manipulations lead to the forms in equations (6.4), i.e

$$x(t) = ((-px_{i-1} + 3px_i - 3px_{i+1} + px_{i+2})/6)t^3 +$$
$$((px_{i-1} - 2px_i + px_{i+1})/2)t^2 +$$
$$((-px_{i-1} + px_{i+1})/2)t +$$
$$(px_{i-1} + 4px_i + px_{i+1})/6$$

$$y(t) = ((-py_{i-1} + 3py_i - 3py_{i+1} + py_{i+2})/6)t^3 +$$
$$((py_{i-1} - 2py_i + py_{i+1})/2)t^2 +$$
$$((-py_{i-1} + py_{i+1})/2)t +$$
$$(py_{i-1} + 4py_i + py_{i+1})/6$$

By identification with (6.4), we find that

$$a_x = (-px_{i-1} + 3px_i - 3px_{i+1} + px_{i+2})/6$$
$$b_x = (px_{i-1} - 2px_i + px_{i+1})/2$$
$$c_x = (-px_{i-1} + px_{i+1})/2$$
$$d_x = (px_{i-1} + 4px_i + px_{i+1})/6$$

and

$$a_y = (-py_{i-1} + 3py_i - 3py_{i+1} + py_{i+2})/6$$
$$b_y = (py_{i-1} - 2py_i + py_{i+1})/2$$
$$c_y = (-py_{i-1} + py_{i+1})/2$$
$$d_y = (py_{i-1} + 4py_i + py_{i+1})/6$$

Equation (6.5) is then used to estimate the curvature at a given pen position P_i. Looking at the above expressions for coefficients c_x, c_y, b_x, b_y, the curvature at the pen position, depends only on two neigbouring points P_{i-1} and P_{i+1}; and is

$$C_{pi} = \frac{4((px_{i+1} - px_{i-1})(py_{i-1} - 2py_i + py_{i+1}) - (py_{i+1} - py_{i-1})(px_{i-1} - 2px_i + px_{i+1}))}{((px_{i+1} - px_{i-1})^2 + (py_{i+1} - py_{i-1})^2)^{3/2}} \quad (6.6)$$

6.2.3.1 Strategy for high level partitioning points

Equation (6.6) shows that the curvature is expressed in terms of first and second differences, which may be noisier, if the original data points are noisy; to reduce the noise problem some kind of smoothing must be applied.

This leads us to the question; how is the smoothing to be carried out ?

The displacement between the original pen position and the approximated one produced from the approximating B-spline is used to evaluate the importance of the point; i.e whether it is a high level partitioning point or not. Considering the expressions developed above for x(t) and y(t) we can see that a displacement of a given pen position P_i from this cubic B-spline description is

$$\partial_x = d_x - px_i = px_{i-1}/6 - px_i/3 + px_{i+1}/6 \quad (6.7)$$

$$\partial_y = d_y - py_i = py_{i-1}/6 - py_i/3 + py_{i+1}/6$$

To evaluate the importance of the point P_i, the curvature is recomputed by fitting a cubic B-spline to the displaced points. By displaced points, we mean that, the points P_{i-1}, P_i, P_{i+1}, which are involved in the estimation of curvature are translated by $+\partial$, this means that through translation

P_{i-1} becomes $Pt_{i-1} = P_{i-1} + \partial$; i.e $P_{i-2}/6 + 2P_{i-1}/3 + P_i/6$

P_i becomes $Pt_i = P_i + \partial$; i.e $P_{i-1}/6 + 2P_i/3 + P_{i+1}/6$

P_{i+1} becomes $Pt_{i+1} = P_{i+1} + \partial$; i.e $P_i/6 + 2P_{i+1}/3 + P_{i+2}/6$

The curvature becomes

$$C_t = 2\,(c_{tx}b_{ty} - c_{ty}b_{tx}) / (\,c_{tx}^2 + c_{ty}^2\,)^{3/2} \quad (6.8)$$

where

$c_{tx} = (-pxt_{i-1} + pxt_{i+1})/2$

$c_{ty} = (-pyt_{i-1} + pyt_{i+1})/2$

$b_{tx} = (pxt_{i-1} - 2pxt_i + pxt_{i+1})/2$

$b_{ty} = (pyt_{i-1} - 2pyt_i + pyt_{i+1})/2$

Substituting the translated points into (6.7) lead to the second displacement $s\partial$ ($s\partial x$, $s\partial y$) and

$$s\partial x = px_{i-2}/36 + px_{i-1}/18 - px_i/6 + px_{i+1}/18 + px_{i+2}/36 \quad (6.9)$$

$$s\partial y = py_{i-2}/36 + py_{i-1}/18 - py_i/6 + py_{i+1}/18 + py_{i+2}/36$$

Now the total displacement is $a\partial = \partial + s\partial$; adding (6.7) and (6.8) results in

$$a\partial x = px_{i-2}/36 + 2px_{i-1}/9 - px_i/2 + 2px_{i+1}/9 - px_{i+2}/36$$

$$a\partial y = py_{i-2}/36 + 2py_{i-1}/9 - py_i/2 + 2py_{i+1}/9 - py_{i+2}/36$$

Considering equation (6.8), we can see that the estimate of the curvature at point P_i is a moving weighted average of 5 successive points; i.e 2 points preceeding the point of interest and 2 points after it. To declare a point as high level partitioning point, the following conditions must be fullfilled:

a. The total displacement a∂ is larger than a given threshold ∂_h

b. The estimate of the curvature C_t (equation 6.8) is larger than a given threshold c_h.

c. The curvature C_t must be a local maximum.

Pratical results

Figure 6.6 and 6.7 show the results obtained by the method described in this section. Here the same pictures are magnified to ensure that the true corners are detected easily. In all the pictures, the curvature threshold c_h was set 0.6, and the displacement threshold $\partial h = 0.3$. The number of detected corners is smaller than the one from the k-curvature; in the sense that no groups of close corners are found, thus there is no further processing to extract the real corners. True corners are effectively detected. The method is not iterative, this is the reason why the average computation time is 5 seconds, this is 1.4 times faster than the 5-curvature method discussed in the previous section.

In terms of effectiveness, the performance of method 2 is better than method 1, thus we have adopted as our high level partitioning technique, so the low level segmentation discussed in the next section assumes that the corner detection is performed by method 2.

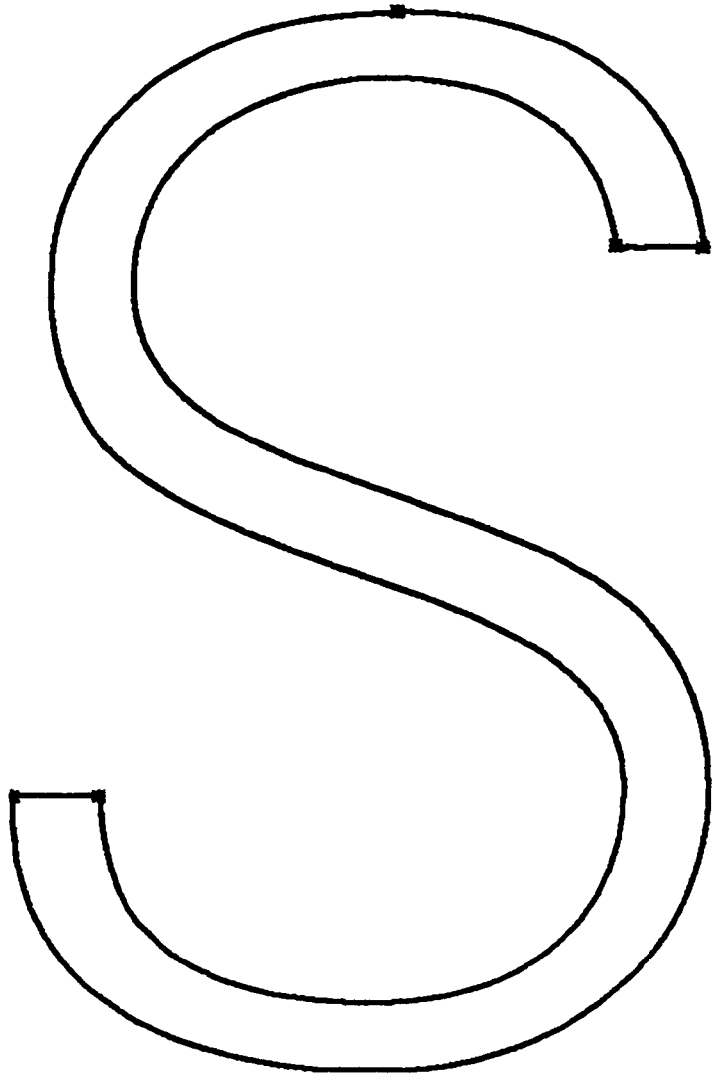


Fig.6.6
Higher level segmentation of test trace
by partitioning method 2.
Curvature threshold = 0.6
Displacement threshold = 0.3

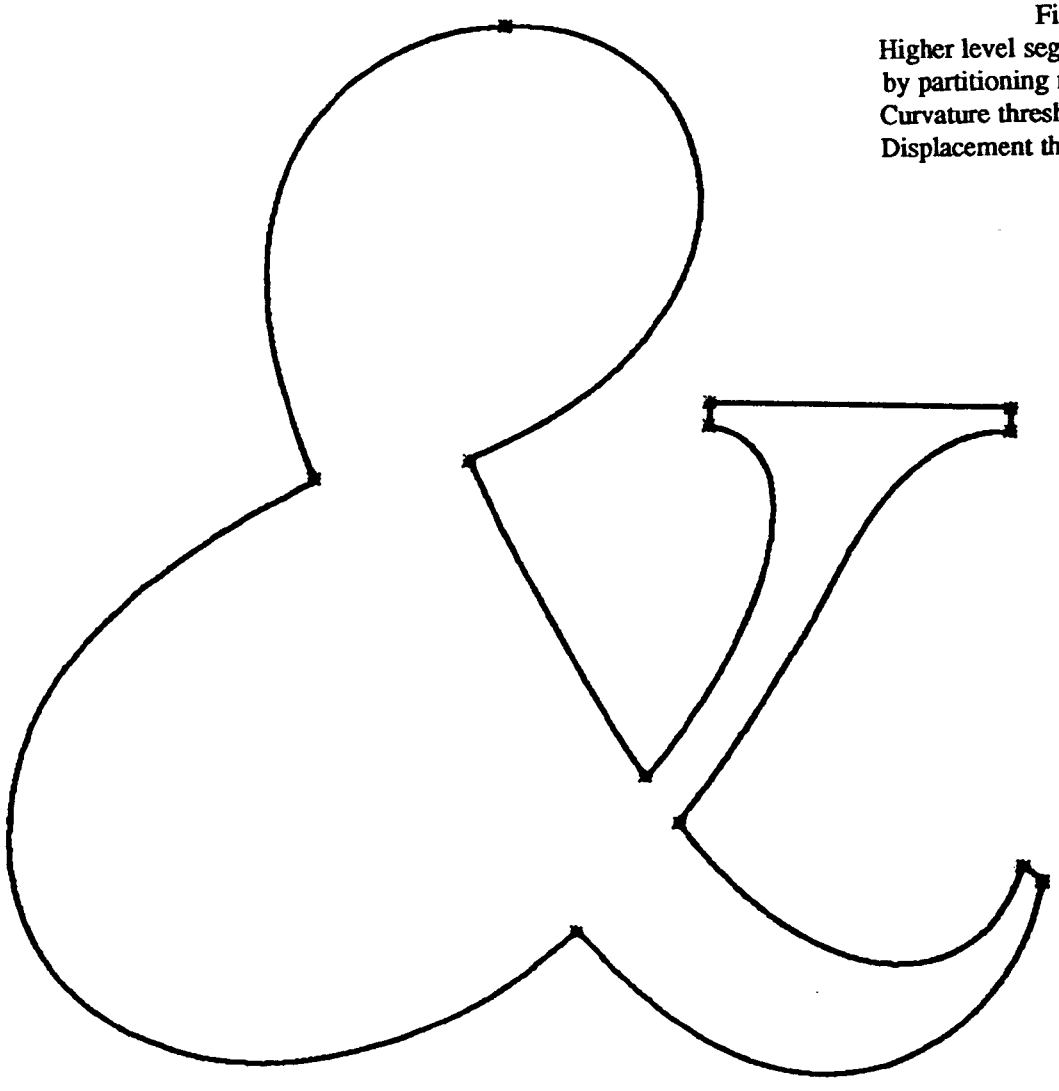


Fig.6.7
Higher level segmentation of test trace
by partitioning method 2.
Curvature threshold = 0.6
Displacement threshold = 0.3

6.3.Low level segmentation of pen traces partitioned by the high level partitioning technique discussed in section 6.2.2

In this section, we aim to represent each high level segment by a sequence of small segments, subject to the condition that the resulting polygon preserves the shape and the size of the original segment, within a given tolerance. The advantage of such a procedure is that it allows a high level segment under consideration to be represented by a smaller number of points; namely the vertices of the approximating polygon, thus achieving noise reduction and data compression.

In chapter 5, the algorithms discussed share one disadvantage: the degree of approximation is controlled by specifying the value of tolerance parameter, which varies with algorithm choice, in a suitable way depending on the curve representation scale. Such a scale dependent tolerance value gives an absolute, (as opposite to relative) measure of the maximum tolerated mismatch between data points of actual curve and segments approximating it. It does not permit us to take into account the relative importance that a given absolute mismatch has in the context in which it arises. Moreover, since in the use of hand generated material there are likely to be pen traces of different sizes, the adequate approximation of all traces requires the specification of several tolerance values instead of just one. As an example, consider the two traces of Fig.6.8, consisting of two circular shapes of the same significance, but of different relative size. The same tolerance value used for approximating both traces would cause an unacceptable shape distortion of the smallest one. This can be appreciated in Fig.6.9 where the inner circular shape is unrecognizable, whereas the outer shape clearly conveys the original form (i.e circular). We can see that the outer shape is over represented; a slightly higher tolerance would still produce an acceptable result for the outer shape, but the inner shape would become worse and convey the unintended pictorial information; this is shown in

Fig.6.10. For this reason, it is felt that an algorithm based on a scale independent tolerance parameter, in the sense that its values specify the maximum allowed percentage variation of some suitable feature of the polygon itself is needed.

So allowing a scale independent control of approximation is the aim of the algorithm to be discussed in this section. It is of merging type (PAVL82, DANIEL83), and sequentially performs polygonal approximation. The approximation is based on a tolerance expressed in terms of percentage of the area enclosed by the pen trace segment under consideration.

In our following discussions, we assume that each high level segment is closed, to simulate this, we join its starting point to its ending point by a virtual straight line. As we have an closed segment we can calculate the enclosed area. Now let us analyze the algorithm:

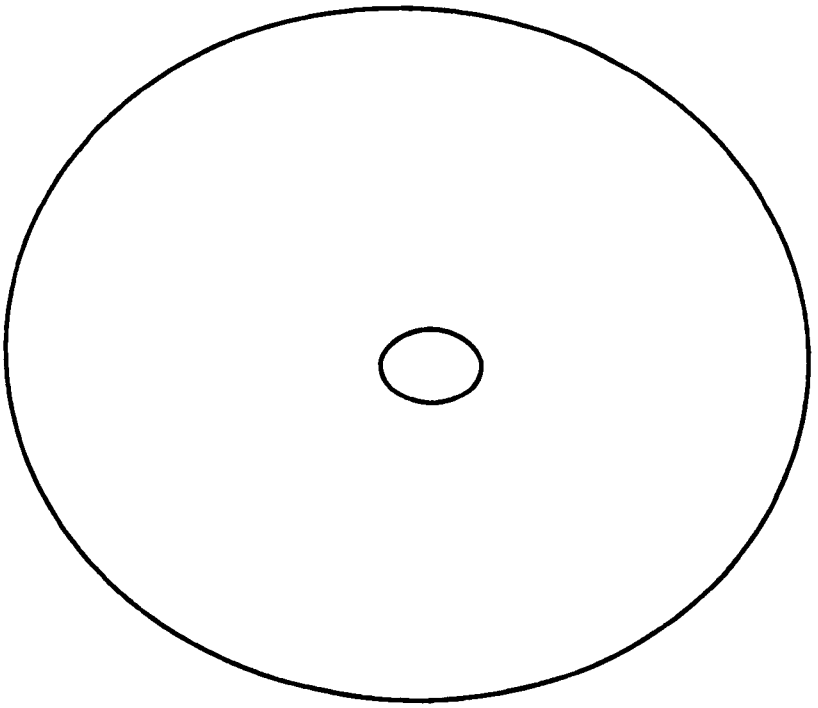


Figure 6.8 Original pen traces

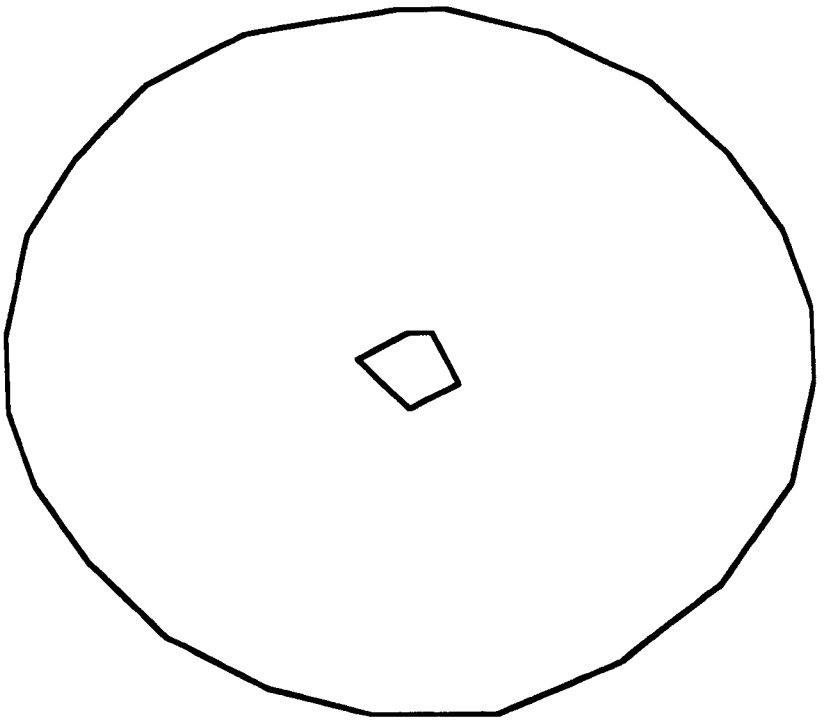


Figure 6.9. polygonal approximation

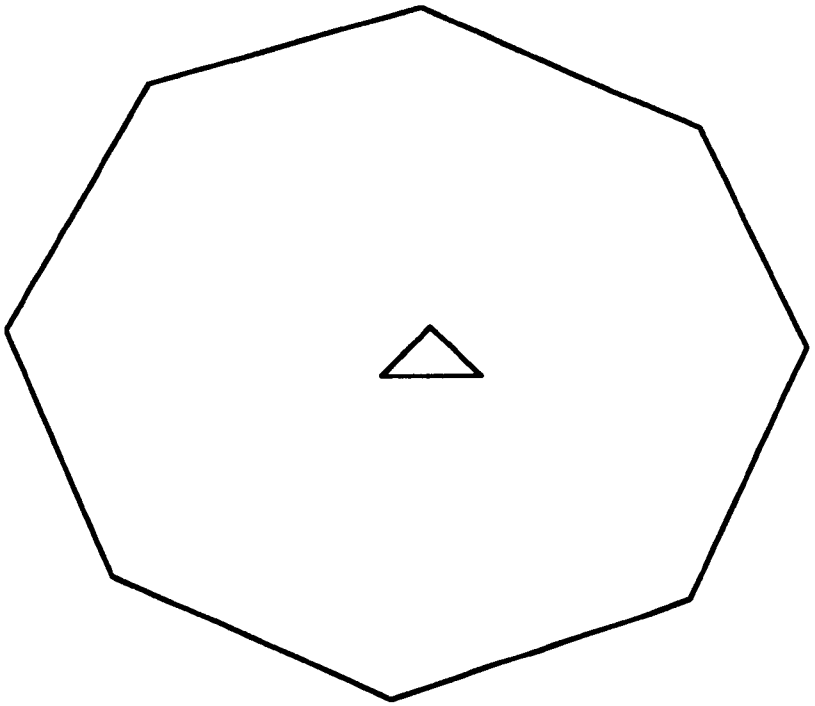


Figure 6.10 polygonal approximation

6.3.1 Algorithm analysis

The algorithm is iterative and is based on a scale independent approximation criterion. The degree of approximation is determined by the required tolerance. The tolerance is given as a percentage error introduced by the approximating process on the polygon area, consequently, the algorithm is area preserving. For this reason, we think that the algorithm is mostly suitable for processing closed pen trace segment, this puts a weight on our above assumption.

The choice of the area as a reference feature is justified by the following considerations :

Consideration 1: The ratio of the area bounded by a segment to that bounded by the convex hull of one of its inflexions is an effective criterion to decide whether the inflexion represents a meaningful or a negligible shape characteristic.

Consideration 2: The enclosed area is a significant feature of closed curves in artistically hand generated kinds of work (e.g typography COUEGN81 , KAROW87); hence, in this respect, an area preserving approximation seems more reliable than others which do not control nor estimate the area variation of the relevant polygons.

Consideration 3: Area preservation also guarantees shape preservation, under the condition that the tolerance is homogeneously distributed along the polygon boundary.

Let S be a high level segment (thought of as closed) and suppose the pen positions P_i (x_i , y_i) and the area A_0 of an initial polygon approximating the segment are given. The vertices of such a polygon may be either the pixels of the digital segment itself or a subset of them, obtained, for instance, by deleting all intermediate pen positions in each sequence of collinear ones. Let E_0 be an a priori estimate of the maximum percentage area variation which can be tolerated, and $E = E_0A_0$ the actual tolerable variation.

E is referred to as the global tolerance. A local tolerance value to be used when locally evaluating the approximation error is the error introduced when trying to merge adjacent pen positions. This can be obtained by homogeneously distributing the global tolerance value along the bound of the high level segment under consideration. The homogeneity criterion adopted is critical in order to preserve shape features. For this purpose it appears convenient, before evaluating the local tolerance, to delete all the polygon vertices that are intermediate in a sequence of collinear ones. Besides speeding up computations, this avoids smoothing of essential features of the pen trace.

We know that a polygon boundary is characterized by both side number and perimeter length, thus there at least two reasonable strategies for evaluating a local tolerance.

Strategy one:

The first strategy, based on side number, approximates more strongly sequences of many short sides. It evaluates the unitary tolerance associated with all the N sides of the segment under consideration. It evaluates the unitary tolerance, associated with every of the N sides of the polygon, as

$$E_{u,1} = E/N = (A_0E_0)/N \quad (6.10)$$

so the tolerance to be used when trying to merge k sides is simply obtained by multiplying (6.10) by k, i.e

$$E_{k,1} = kE_{u,1} \quad (6.11)$$

This is essentially adding unitary tolerance expressed in (6.10), thus

$$E_{k,1} = E_{k-1,1} + E_{u,1} \quad (6.12)$$

Strategy two:

The second strategy performs the tolerance subdivision according to the length, thus producing a stronger approximation on sequences of longer sides. It computes the tolerance per unit length of the boundary as:

$$E_{u,2} = (A_0 E_0) / P \qquad (6.13)$$

The tolerance for k sides is then obtained as

$$E_{k,2} = (E_{u,2} L_T) \qquad (6.14)$$

where $L_T = \Sigma L_i$ (i varies from 1 to k), approximates the length of the segment after the pen has moved k times from a given initial position. We can see that

$$E_{k,2} = E_{k-1,2} + E_{u,2} L_k \qquad (6.15)$$

where L_k is the length of kth side.

Strategy three:

This last strategy evaluates the local tolerance as an average of the two previous ones, and gives better results in practice, presumably because it averages the performances of the two previous approaches, i.e

$$E_{k,3} = (E_{k,1} + E_{k,2})/2 \qquad (6.16)$$

It can be seen that the sum of the local tolerances employed during approximation is, in all cases, never greater than the global tolerance.

For dimensional consistency with the proposed tolerance parameter, the local error due to approximation must be evaluated as area variation. It is computed as the absolute value of the area of the polygon bounded by a sequence of sides and by the segment approximating them, that is, joining the extreme vertices of the approximated sequence of points; this is illustrated in Fig.6.11. Since the polygon with vertices P_i (x_i, y_i) is often not simple, areas are evaluated with their signs, as in DANIEL83.

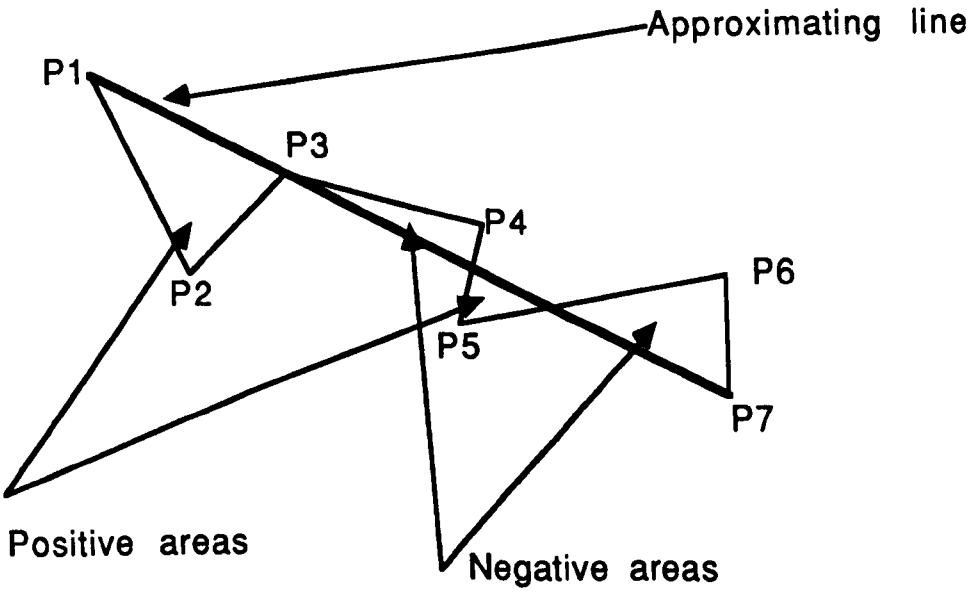


Fig.6.11 Area computation.

$$AE_k = \Sigma\, (x_{i+1} - x_i)*(y_{i+2} - y_i) - (x_{i+2} - x_i)*(y_{i+1} - y_i)$$

so that the sum of all local errors actually corresponds to the area variation of the polygon. It must be pointed out that the local area error, as well as the local tolerance, can be computed iteratively by simply updating, each time a new point is considered, the values computed at the previous step; this approach gives the algorithm a linear time complexity.

The algorithm processes a high level segment made of N pen positions in a sequential manner. The computational complexity of the algorithm is O(N). During the analysis, we mention the preliminary deletion of successive points lying on a straight line, one may think that it does make the

algorithm time consuming, but in fact it does not increase the complexity of the algorithm, since it can be achieved in linear time
(i.e processing time is a linear function of the number of points) simultaneously to forming the input list of the pen positions.

To recapitulate, the algorithm is based on scale independent, non-dimensional tolerance, representing the allowed percentage area change of polygons to be approximated. Due to this choice, a single tolerance value also produces acceptable results when applied to different size curves of the same shape. Moreover, the relative area change introduced by the approximating process can be controlled, while preserving the essential shape characteristics.
The algorithm is sequential, hence its results are dependent on the initial point. The algorithm is fast, and for a given segment, the processing time is a linear function of the number points N.

Pratical results

Strategy three (6.16) was used for experimental purposes because it produced better results. Applying this algorithm on Fig.6.5 leads to Fig.6.12 and Fig.6.13 for respective tolerances E_0 = 3 % and 6 %. The high level partitioning points are in red, whereas the small circles represent the sampled points produced by the above algorithm. Initially Fig.6.5 requires 4271 points. Applying this algorithm requires only 50 points, whereas the most efficient method of chapter 5 required 83 points. Here we gain in data compression, but the price which has to be paid is the long computing times, which obviously prohibit the algorithm from being adopted for real time applications.

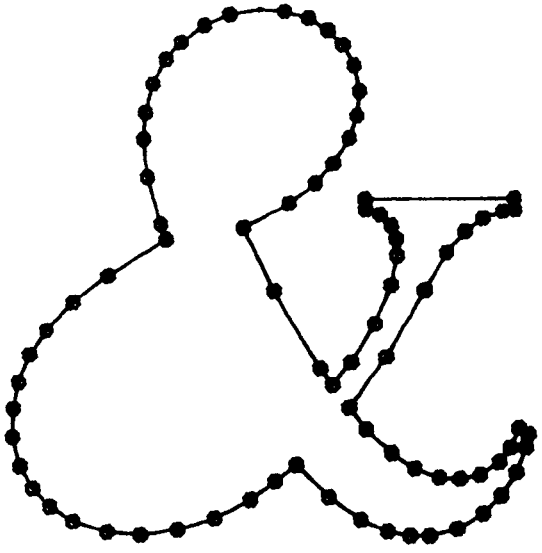


Fig. 6.12
Low level partitioning of test trace (Fig.6.5)
Tolerance E = 3 %

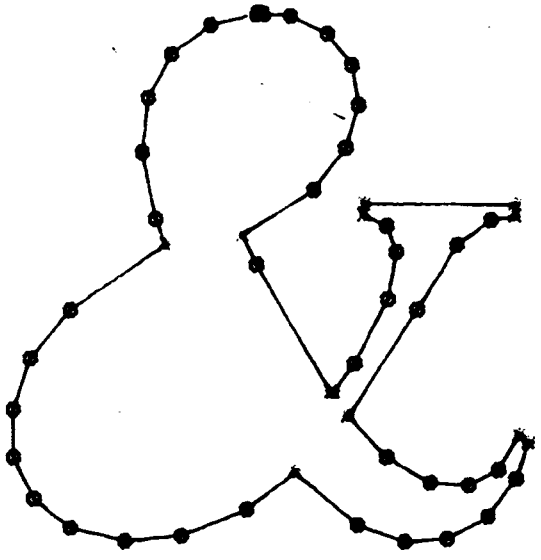


Fig. 6.13
Low level partitioning of test trace (Fig.6.5)
Tolerance E = 6 %

Similar results were obtained for Fig.6.4. This is shown in Fig.6.14 and Fig.6.15, where the tolerances were again 3% and 6% of the areas of the supposed closed high level segments respectively .

Fig.6.4 requires 4384 points. The most efficient real time approximation of chapter 5 produces 101 points; but the present non real time algorithm claims only 42 points, and the shape is genuinely preserved.

On average, from the point of view of data compression the algorithms described in this chapter are twice as efficient as the most efficient algorithm discussed in the previous chapter. But the computation times are very much higher. How much higher depends upon three criteria:

1. The material to be processed.
2. The computing technology used.
3. The coding of the algorithms in terms of the efficiency of the computer language used and the ability of the programmer to code the algorithm efficiently in the specified language.

The need to know all pen positions in advance of applying the algorithm discussed in this chapter indicates that it cannot be applied in real time; unless the delay for each segment is acceptable.

6.4 Looking forward to higher order polynomial approximation.

In the last three chapters, including this one, the geometric primitive used to reconstruct the trajectories of the pen has been the straight line. In some hand generated material, such as typographic characters or any artistic work produced with a light pen, the simple straight approximation may not be good enough for the following reasons :

Fig. 6.14 ce
Low level partitioning of test trace
Tolerance E = 3 %

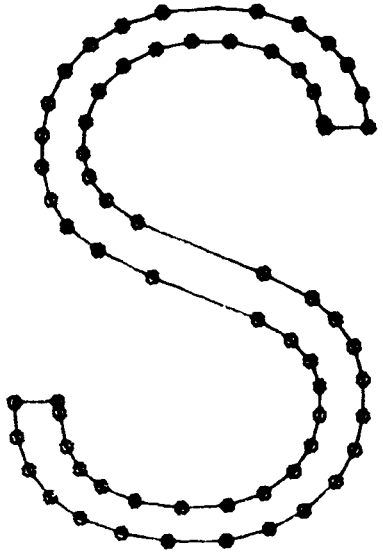
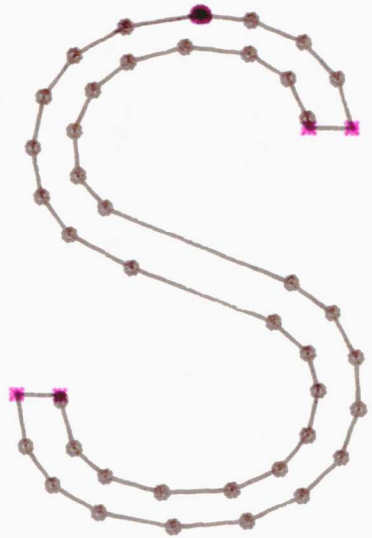


Fig. 6.15
Low level partitioning of test trace (Fig.6.4)
Tolerance E = 6 %



1. Aethestic quality of the material may be lost.
2. Assuming scale independence, a linear approximation may produce too many lines for curved regions.
3. Given a material described by straight lines, a magnified version of the material will undoubtedly show slope and curvature discontinuities between successive straight lines.

What is said here is only valid for curved regions of a material. However for straight parts of the material, genuine straight lines should be provided.

The magnified versions of Fig.6.13 and Fig.6.15 are shown in Fig.6.16 and Fig.6.17. We can clearly see that the discontinuities are accentuated.

A curved segment should be regenerated by linking together a chain of low level segments which merge into one another at least, tangentially. Visually and graphically speaking, this would have the advantage that the curved path would be smoothed out in steps, and would smooth local irregularities because of the same gradient and curvature. A simple straight line approximation cannot cater for these requirements, therefore in the next chapter, we shall investigate into the higher order polynomial approximation of the pen traces.

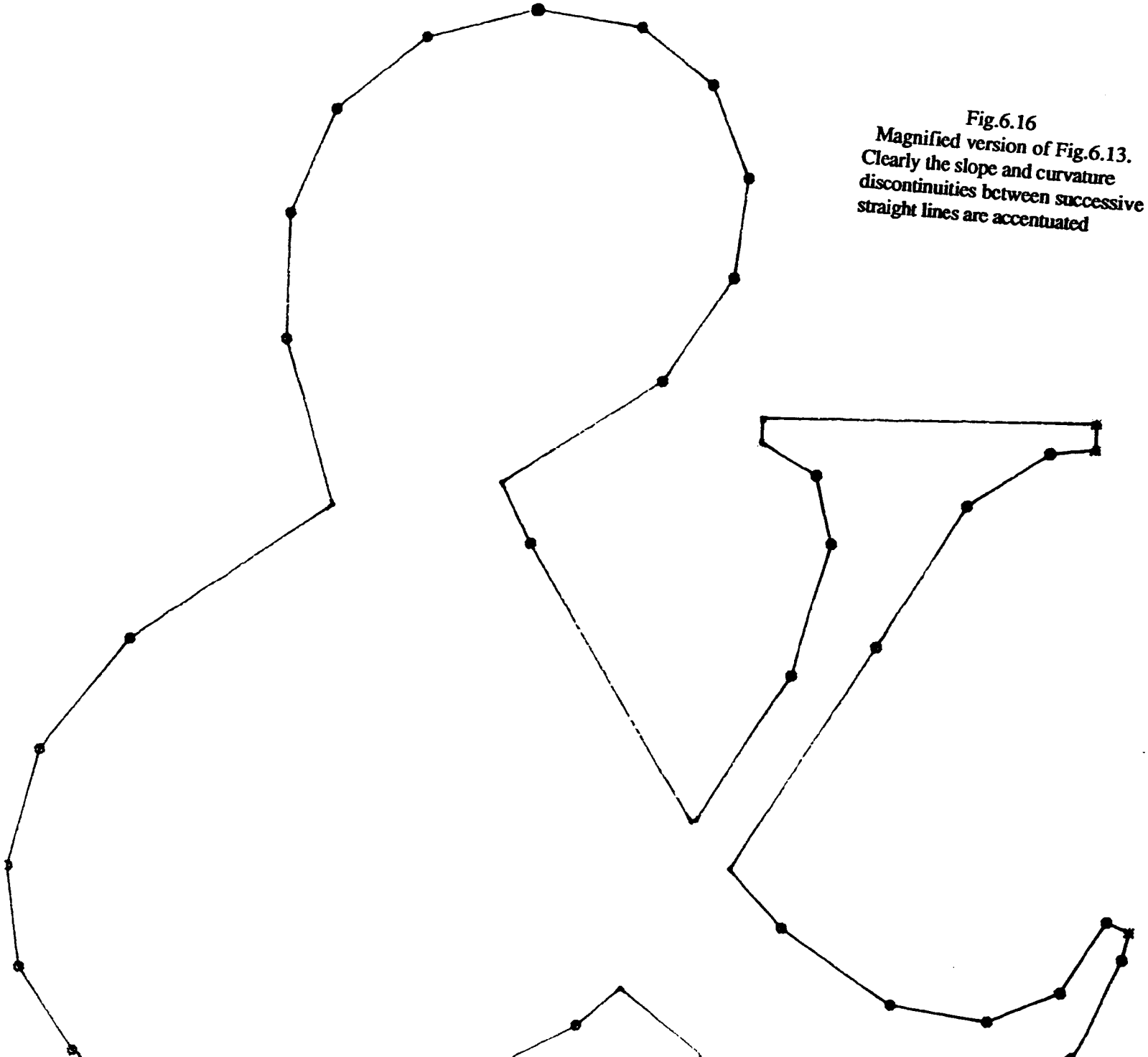


Fig.6.16
Magnified version of Fig.6.13. Clearly the slope and curvature discontinuities between successive straight lines are accentuated

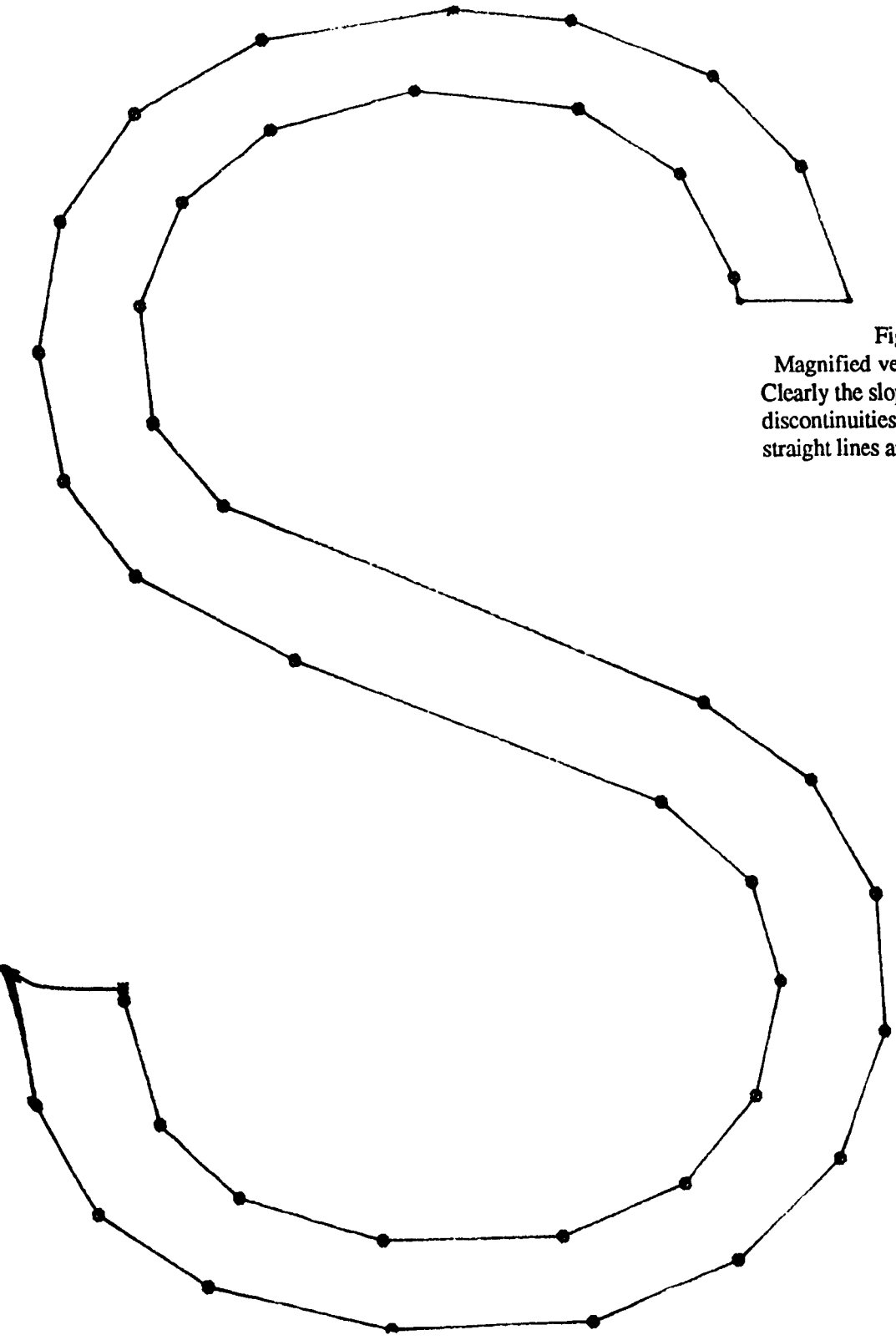


Fig.6.17
Magnified version of Fig.6.15. Clearly the slope and curvature discontinuities between successive straight lines are accentuated

6.5 Entropy results

The probability distributions of Δx and Δy were measured and used to estimate the entropy rate of the signal. The results are presented in the following table where the average sampling rate was about 8 Hz for 6 % tolerance (see section 6.3 above)

n	H_n (bits)	Theoretical bit rates (bits /s)	practical limiting bit rates (bits /s)
0	15	120	127
1	11	88	96
2	10	80	89
3	9.23	74	86
4	8.73	70	79
5	7.83	63	75
6	6.43	52	65
7	5.85	47	60
8	5.02	40	55

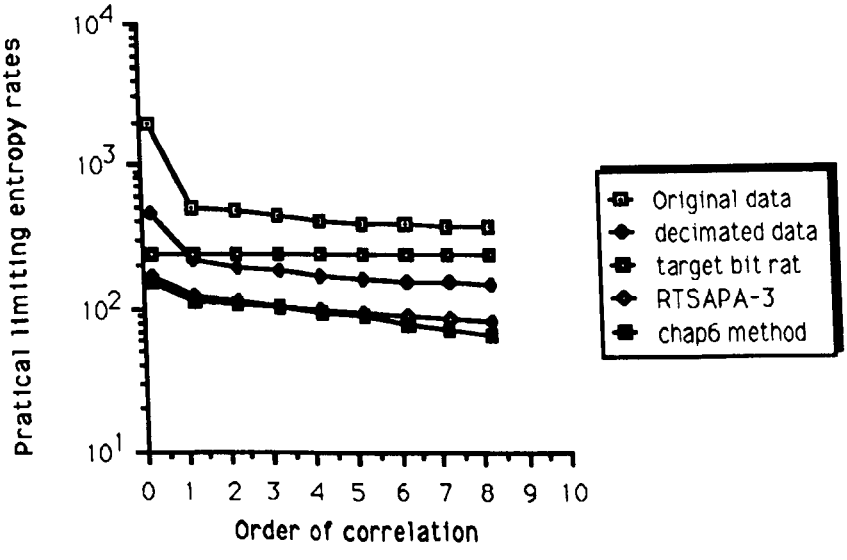
Theoretical and practical limiting bit rates.

A study of the above table indicates that :

1. The lowest limiting entropy rate estimate is 55 bits/s is associated with 8th order correlation between successive vectors. Compared with the results of RTSAPA-3 (n = 8 ; Table 5.6 of the previous chapter), the drop in limiting entropy rate is over 21 %.
2. The higher the order of the correlation, the lower the entropy per signal element.
3. The entropy rate of the signal decreases with higher order correlation.

6.6 Graphical comparison of entropy rates of the signals, estimated from original and reduced data.

As in the previous chapters, we present the entropy measurements of chapter three to the present chapter, in the same graph for visual evaluation. The following figure, clearly shows that the entropy results obtained from the method of this chapter, are much more efficient than the results obtained from the methods discussed in the previous chapters.



The bottom curve of the above figure is the graphical presentation of the practical limit entropy rates estimated from the relative frequency distributions of $\Delta x / \Delta y$ produced by the approximating method discussed in this chapter.

6.7 Conclusions

This chapter has shown that, in the event of off line transmission of hand generated graphic material, efficient algorithms could be used. Here, off line processing requires knowledge of the spatial information about traces before hand, i.e an a priori knowledge of points making various traces of a hand generated material.

The efficiency here means data compression efficiency. A shape preservation algorithm has been discussed, and when combined with our suitable corner detection algorithm, it has yielded highly compressed pictures (i.e in a storage space saving manner).

The entropy rate estimates are lower than the estimates from the previous chapters. This is explained by the fact that, we have a much improved data reduction, which in turn, leads to a lower average sampling rate and a lower bandwidth.

It is thought that a higher order (at least degree two) polynomial description of the trajectories of the pen should combat the problems of kinks which are associated with the straight line (first order polynomial) approximation. The work described next will tackle the problem of genuine smooth transitions between segments of the segments of the pen trajectories.

7. PIECEWISE POLYNOMIAL DESCRIPTIONS OF PEN TRACES

Throughout chapters 4 to 6, the interpolator used to reconstruct the trajectories of the pen, has been the straight line. When only legibility matters, the straight line interpolation is acceptable, if it conveys the intended message. Assuming that the hand generated material is the work of an artist; the aethestic "look" of the regenerated material, might be one of the overriding reconstruction requirements; should this be the case, straight line fitting alone is not the answer. In general, a pen trace is a combination of straight parts and curved parts. So where, a straight line is needed, it should be correctly catered for; but for a curved part, straight line piecewise description does not provide a smooth transition between pieces of a curved part of the pen trace under consideration. It is thought that a piecewise higher order polynomial interpolator, with the following characteristics

1. Point continuity.
2. Tangent continuity in direction, not necessarily in magnitude, (for non corner points).
3. Curvature continuity (for non corner points).

where needed, could be a suitable alternative to straight line only interpolation which has been used so far.

7.1 Introduction

In our work, so far, the geometric primitive used for hand generated material description has been the straight line. The technique used is to segment the pen trace for coding, using the breakpoints of the segmentation as the data points that are used for reconstruction. An interpolator, a curve that passes through the selected points is then used to reconstruct the picture. In this chapter, instead of using the straight line as an interpolator, we shall investigate into approximating methods which

cater for at least gradient continuity when pieces of the curved part of a pen trace, are sequentially linked. The remainder of the chapter is structured as follows :

1. Brief theory and techniques for polynomial approximations.
2. Application of Bezier method.
3. Application of B-spline method.
4. Conclusions.

7.2 Approximation problem.

The approximating functions we use are piecewise polynomials. Because hand generated material is made of complex curves, global functions, i.e functions defined over an entire domain should be avoided. AKIMA70 gives examples of a curve fitted by a polynomial, Fourier cosine series, cubic spline and some other (osculatory) piecewise cubic polynomials which differ from each other only in the numerical computation of slopes. AKIMA70 's experiments indicate that piecewise polynomials are more suitable. Also, it is important to observe that a careful choice of a derivative computation may lead to better approximations. Another indication of the usefulness of piecewise polynomials, especially piecewise cubic polynomials, is that many people are using them, as can be judged from the large literature in this area (BRODLIE80). YAMA78 uses cubic B-splines, for representing a hand-drawn curve, but his knot (i.e interpolating point) point selection is only semi-automatic, in a sense that he uses computer aided design interactive tools to choose knots; e.g his method requires the interactive user to select knots on the curve and to indicate when more knots are necessary to achieve the required accuracy; his technique is interactive and non-adaptive, whereas we wish to find an automatic and adaptive algorithm.

We will not discuss the theory of splines, because it can be found in many textbooks (e.g DEBOOR78); however, where appropriate and necessary, some basic fundamental results will be recalled.

If we wish to describe our hand generated material by analytic forms; i.e mathematical models, the literature on mathematical methods offers us two kinds of of representations classified as being non-parametric or parametric (ROGER76).

A representation of a curve in a two dimensional space is non-parametric if it is expressed in the implicit form $f(x,y) = 0$.

The definition of a line as $y = mx + b$ is non parametric, as is the definition of a circle $x^2 + y^2 = r^2$. In almost all non parametric representations, the explicit form $y = f(x)$ (if it exists) is usually adopted because it is, in general simpler and hence easier to manipulate.

In a parametric representation, a free independent variable, called the parametric variable, is employed. The original curve is represented as a set of independent functions in this new variable. The curve above would be represented parametrically as $x = x(s)$ and $y = y(s)$. The parametric variable, s is often restricted to a subrange of values so as to bound the functions (e.g $0.0 \leq s \leq 1.0$). Sometimes, s represents arc length along the curve.

In general, hand generated material, is made up of curves which may not be represented in an explicit form, i.e $y = f(x)$, because in this representation a curve may have infinite derivatives and be multivalued. To circumvent these difficulties, we will restrict further discussion to parametric representations.

This is the most common approach to two dimensional curve representation, and provides great flexibility, i.e the curve may cross itself. Further reasons for using parametric representation are discussed in BRODLIE80, KNUTH79.

As seen in previous chapters, if properly used, a polygonal approximation of pen trajectory cannot go wrong, however AKIMA70, MCLAUGHIN83 show that using polynomials other than linear ones may lead to unexpectedly distorted pictures. We are not implying that fitting only straight lines makes everything rosy; but what we are claiming is that a suitable tolerance will usually result in pictures which imitate the original pictures, whereas certain higher polynomials may lead to considerable distortions due to overshoots.

Curve generation methods which use a polygon to define the curve are appropriate for our application. Two major methods exhibit this characteristic. The first one was developed by Decastleau, and further refined by Pierre Bezier, and has been and still is a major tool for designing the car bodies (e.g French company Renault). The second method is an extension of the Bezier method referred to as the B-spline, and is popular in the shipping industry (PAVL82).

We may ask ourselves, if there is anything new in applying them; indeed we claim that our approach is different from the existing approaches, because the interpolating points are selected automatically by a computer algorithm, whereas existing approaches use the human brain to choose the points which are used to generate the interpolator, in other words CAD tools (interactive) use human operator to choose the knots. Usually the user sees the curve on a screen, by changing and adding or deleting the guiding points, he can then produce a curve which suits his application (GILOI78). In our work guiding points are chosen automatically by an algorithmic procedure, this approach justifies the claim that our contribution to this field, in relation to Bezier technique and B-spline technique applied to hand generated material is relatively new because we have found no relevant published material.

In this chapter, we describe two approaches. The first is to send relevant points only; in this case the interpolator must determine reasonable gradients (in direction and magnitude), and then generate the curve segment. The second method is to work out appropriate gradient estimates at each selected point and send it along with the selected point; the decoder will then use the information to generate the curve segment. Both approaches will be analyzed and compared.

7.3 Bezier technique.

Consider the parametric representation of pen trace segments given by

$x = px(t)$

$y = py(t)$, with $0 \leq t \leq 1$

For n + 1 points denoted as $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, ...,$P_n(x_n, y_n)$, the parametric bezier curve is the vector-valued Berstein polynomial of order n given by

$$P(t) = \sum_{i=0}^{i=n} P_i \emptyset_i(t) \qquad (7.1)$$

where

$$\emptyset_i(t) = (n! / i! (n - i)!) t^i (1 - t)^{n - i} \qquad (7.2)$$

are the basis functions; i varies from 0 to n. P(t) and P_i are respectively the column vectors

$$\begin{matrix} px(t) & x_i \\ py(t) & y_i \end{matrix}$$

P0 and Pn are usually the end points of a bezier segment, where

$P_1, P_2 \ldots P_{n-1}$ are usually called the control of the bezier segment described by the polynomial of degree n expressed in (7.1).

The points $P_0, P_1, P_2 \ldots P_n$ are the vertices of a polygon which uniquely defines the curve shape. Only the first and last vertices actually lie on the curve (i.e P_0 and P_n). Some of the interesting properties of the Bezier curve

are that it lies inside the convex hull of the polygon. Also the curve is tangential to the line segments P_0P_1 and $P_{n-1}P_n$ at P_0 and P_n, respectively. Using (7.2), the basis functions for the case n = 3 are the following :

$Øo(t) = (1 - t)^3$

$Ø1(t) = 3t(1 - t)^2$

$Ø2(t) = 3t^2(1-t)$

$Ø3(t) = t^3$

As it can be seen from these equations, the order of the curve is equal to the number of spans or number of vertices minus 1.

Fig.7.1, Fig.7.2 show examples of cubic curves generated by the above basis functions using different vertices.

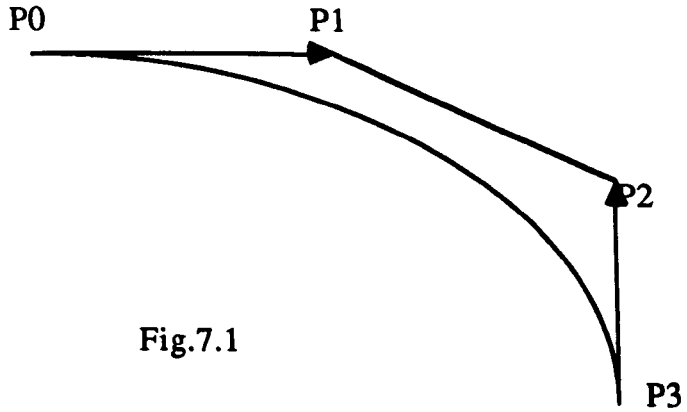


Fig.7.1

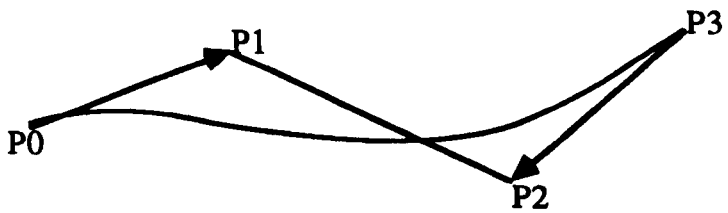


Fig.7.2

From equation (7.1), the first derivatives at the end points are

$p'(0) = n (P_1 - P_0)$

$p'(1) = n (P_n - P_{n-1})$

In the cubic case we have

$p'(0) = 3 (P_1 - P_0)$

$p'(1) = 3 (P_3 - P_2)$

From these equations, we can express the control point P1 and P2 in terms of gradient vectors

$P_1 = P_0 + p'(0)/3$

$P_2 = P_3 - p'(1)/3$

Taking into account the basis functions, the bezier cubic format can be put into standard cubic format as

$$p(t) = P_0 + \mathbf{p'(0)}t + (3(P_3 - P_0) - \mathbf{2p'(0) - p'(1)})t^2 + (\mathbf{p'(0) + p'(1)} - 2(P_3 - P_0))t^3 \quad (7.3)$$

Looking at this expression, we can clearly see that a trace segment which runs from the selected end point P_0 to the selected end point P_3 is described by a cubic which is uniquely defined by the selected points P0 and P3 and the tangent vectors p'(0) and P'(1) respectively at the two ends. Having selected the end points through a segmentation process;
(e.g see chapter 3), **the remaining issue is how do we estimate the tangent vectors associated with the end points ?**
The answer to this question depends upon two approaches :

1. Tangents are estimated from the selected points.
2. Tangents are estimated from the local data during the segmentation process, and sent along with the selected points.

7.3.1 Estimation of tangents from a sequence of selected points

Here, the decision is made to transmit only the automatically selected points. In this paragraph, we present a solution to the interpolation problem: given a set of selected relevant pen locations, with no knowledege about gradient information, use a cubic interpolator to reconstruct the various segments comprised in a pen trace of hand generated material.

To restate the interpolation problem: given an ordered set of selected relevant pen positions, P_i with i varying from 1 to N, generate a curve p(t) such that $p(0) = P_i$ and $p(1) = P_{i+1}$.

On any normalized interval ($0 \leq t \leq 1$), associated with a segment delimited by P_i and P_{i+1}; with reference to equation (7.3) the interpolating curve is expressed as

$$p(t) = P_i + \mathbf{p_i'(0)}t + (3(P_{i+1} - P_i) - 2\mathbf{p_i'(0)} - \mathbf{p_{i+1}'(1)})t^2 + (\mathbf{p_i'(0)} + \mathbf{p_{i+1}'(1)} - 2(P_{i+1} - P_i))t^3 \qquad (7.4)$$

Looking at (7.4) suggests that, all that is required is to obtain local estimates of the gradients $\mathbf{p_i'(0)}$ and $\mathbf{p_{i+1}'(1)}$ using the given selected pen positions.

7.3.1.1 Gradient construction

Our determination of gradients will be dealt with as two dimensional problems. The methods examined are as follows: Midgeley 's method, Bessel method, Akima's method, higher order polynomials.

Midgeley's method

MIDGEL79 suggested the idea of passing a circle through 3 points to determine the slope. Referring to Fig.7.3, the gradient t_v is given by

$$t_v = d_i V_{i-1}/d_{i-1} + d_{i-1} V_i/d_i \qquad (7.5)$$

d_i and d_{i-1} are respectively the distances from P_i to P_{i+1} and P_{i-1} to P_i.

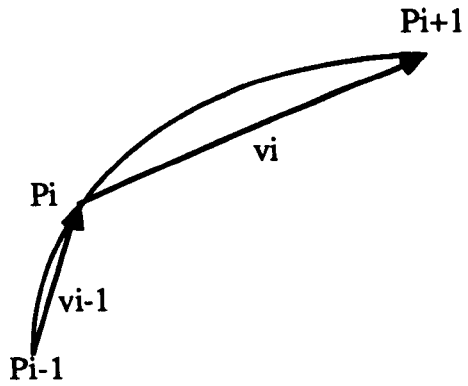


Fig.7.3 Tangent definition for a circular segment

The derivative at the starting point of a trace and end point of a trace were similarly calculated using circular arcs (MIDGEL79, RUTK79).

A problem was associated with the method: the derivative magnitude does not collapse to zero as |Vi + Vi+1| does, as a result the method does not have the ability to embed a straight line segment; moreover it produces reconstructed traces which are too rounded.

Bessel's method

For the bessel interpolator, the slopes are determined by a parabolic (quadratic) fit through 3 points. Given three successive points P_{i-1}, P_i, P_{i+1}; and

d_{i-1} the distance from P_{i-1} to P_i.

d_i the distance from P_i to P_{i+1}

If we use a parabola through these successive points, it can be proved that the gradient at point P_i is (DEBOOR78)

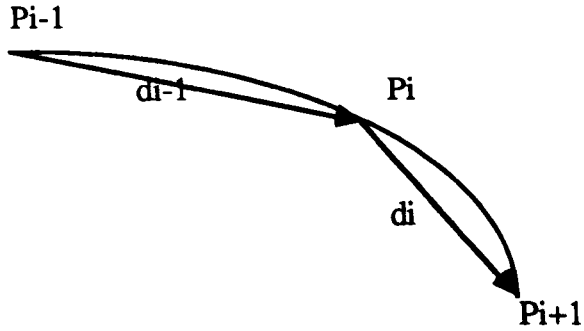


Fig.7.4 Parabolic curve used to calculate the slope at P_i

$$tv = (d_i / (d_{i-1} + d_i)d_{i-1}) (P_i - P_{i-1}) + (d_{i-1} /(d_{i-1} + d_i)d_i)(P_{i+1} - P_i) \quad (7.6)$$

so the components of the gradient vectors are

$$tvx = (d_i / (d_{i-1} + d_i)d_{i-1}) (Px_i - Px_{i-1}) + (d_{i-1} /(d_{i-1} + d_i)d_i)(Px_{i+1} - Px_i)$$
$$tvy = (d_i / (d_{i-1} + d_i)d_{i-1}) (Py_i - Py_{i-1}) + (d_{i-1} /(d_{i-1} + d_i)d_i)(Py_{i+1} - Py_i)$$

So the determination of the slope at a given selected point is a function of its two neighbours (left and right neighbours); a careful look at (7.6) shows that the slope at Pi is a function of slopes of lines connecting P_{i-1} to P_i and P_i to P_{i+1}, which are respectively $(P_i - P_{i-1}) / di\text{-}1$ and $(P_{i+1} - P_i) / di$

Closely spaced points contribute more to the slope value.

Akima's method

AKIM70 introduced a method designed to eliminate the ripples that commonly occur in polynomial or smooth spline interpolation. Considering five successive points P_{i-2} P_{i-1} P_i P_{i+1} P_{i+2}, the slope estimate is given as (AKIM70, DEBOOR78).

$tv = (|m_{i+1} - m_i|m_{i-1} + |m_{i-1} - m_{i-2}|m_i) / (|m_{i+1} - m_i| + |mi-1 - mi-2|)$

where

$m_{i-2} = (P_{i-1} - P_{i-2})/d_{i-2}$

$m_{i-1} = (P_i - P_{i-1})/d_{i-1}$

$m_i = (P_{i+1} - P_i)/d_i$

$m_{i+1} = (P_{i+2} - P_{i+1})/d_{i+1}$

This method eliminates ripples in the interpolating function and is fairly efficient. However, it has not got the ability to embed collinear points. It does not allow closely spaced points to dominate the slope measure. This can be a disadvantage with unevenly spaced data. We have found experimentally that the presence of ripple is not as noticeable in parametric form, as both x(t) and y(t) ripple simultaneously. Akima's method is quite popular (BALL78) and has been the basis for the next method

Renner and Pochop's method

Following Akima's strategy, RENNER82 uses four neighbouring points; two at the left and two at the right of the point of interest, to estimate the slope. Straight line segments are preserved using an approach based on AKIMA70. Considering five consecutive selected points P_{i-2}, P_{i-1}, P_i, P_{i+1}, P_{i+2}, a condition to determine the slope tv at Pi is sought (refer to Fig.7.5). It seems reasonable to require tv to approach V_{i-1} if V_{i-2} approaches V_{i-1}, and similarly, it should approach V_i if V_{i+1} approaches V_i, where $V_i = P_{i+1} - P_i$. On the basis of this requirement, the slope tv will be defined as the linear interpolator of Vi-1 and Vi as follows:

$$tv = (1 - ß)V_{i-1} + ßV_i \qquad (7.8)$$

where ß = B / (B + C)

$B = |V_{i-2} \times V_{i-1}|$; $C = |V_i \times V_{i+1}|$

B and C are the areas of the parallelograms spanned by V_{i-2} and V_{i-1}, and V_i and V_{i+1} respectively.

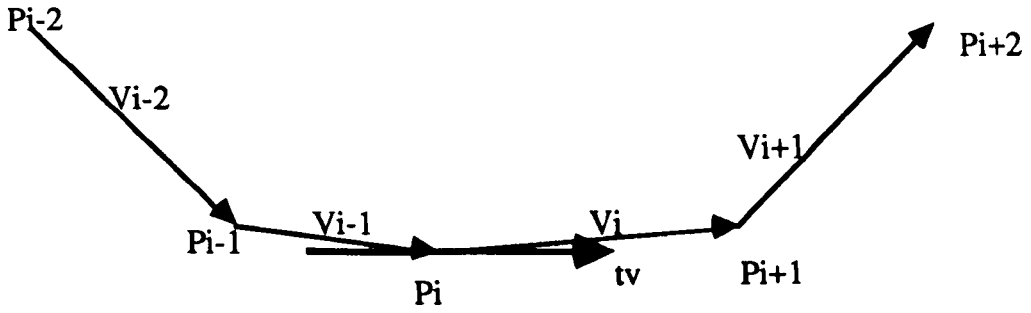


Fig.7.5 Definition of the tangent at P_i (Renner's method)

Looking at the directions of the differences vectors, we see that the following special cases deserve attention.

conditions B = 0 ; C ≠ 0 imply that Vi-2 and Vi are collinear, as a result ß = 0 and tv = V_{i-1}

Conditions B ≠ 0 and C = 0 imply that V_i and V_i =1 are collinear, thus tv = V_i

Conditions B = 0 and C = 0 lead to two cases

a. Vi-2 and Vi-1 as well as Vi and Vi+1 are collinear and the slope is defined as

$$tv = (V_{i-1} + V_i)/2$$

b. All the vectors V_{i-2}, V_{i-1}, V_i and V_{i+1} are collinear, and the slope is defined as tv = V_i

tv depends on the direction of the difference vectors and on their magnitudes as well. If , for example we let $|V_{i-2}|$ tend to infinity, then tv will approach Vi; that is, the growth of A pulls the curve towards the cord P_iP_{i+1}. Similarly, an increase in B pulls the curve towards $P_{i-1}P_i$.

A very desirable property of the slope definition above is that the interpolator

can reproduce straight lines through three consecutive collinear data points, since the tangents are collinear too. This means that the interpolator does not suffer from the unwanted wiggles and oscillations in the part of the curve where collinear data points appear.

Higher order polynomials

Using a higher order (nth order), polynomial through n points is a simple extension of the Bessel interpolator. No added accuracy is gained, since the interpolator is cubic. Because symmetry is desirable in the coding of drawings, only even polynomials need be considered.

Unfortunately, the formulation for even the quartic becomes too complicated, involving 5 products in the weighting factor of each term (DEBOOR78).

The formulation for higher order polynomials is even more complex, for this reason, this method was not further investigated.

Comparisons of slope derivation methods.

All methods examined, show that the slope (tv at point i) is a scalar weighted sum of the sample points at point i = $\Sigma w_{ij}P_j$ with j covering the number of points involved in the estimation of the slope. The weights w_{ij} depend on the distances separating the points. All the method are local, this means that a segment of the curve between two given points is determined only by points in a certain range of this segment and the points beyond this have no influence on the segment shape. Consequently, the shape of any data points changes the curve only in the neighbourhood of this point, leaving other parts unchanged. This property is very important in our type of work, because we are concerned with a step by step reconstruction of the pen traces, and we often wish to alter one section, without causing any change in the rest of the curve.

The circular fit is not suitable for drawing reconstruction. The pictures it produces are too rounded. The Bessel interpolator has the highest accuracy and precision. It is quite simple, but does not smooth as well as the other methods that use more of the neighbouring points. It is to be preferred in those one dimensional applications where accuracy is a primary consideration (DEBOOR78). Akima's method is very useful for ripple elimination in one dimension, but has no advantages in the two dimensional parametric case. The higher order polynomials may perform better because of the increased number of points considered. However the computational complexity involve preclude their being used.

Renner and Pochops method has proved to be effective, because not only it has the ability to reproduce intended straight parts of the curve, but it reflects the geometric characteristics of the points.

The shape of the resulting curve is essentially influenced by the magnitude of the tangent vectors. If we select unsuitable tangent vectors, we can obtain rather odd results. The expressions for slope yield the direction; not the magnitude of the gradient vectors. One can derive a unit tangent vector for each one of the above expression; having got the unit vector; but the actual magnitude used was obtained from MANN72, FORREST68, who worked out a suitable magnitude for weighting the unit vector as:

$$2D \,/\, (1 + \mu\cos\phi_{i+1} + (1 - \mu)\cos\phi_i) \text{ for gradient at point } P_i \qquad (7.6)$$

$$2D \,/\, (1 + \mu\cos\phi_i + (1 - \mu)\cos\phi_{i+1}) \text{ for gradient at point } P_{i+1}$$

The parameters of these expressions are shown in Fig.7.6

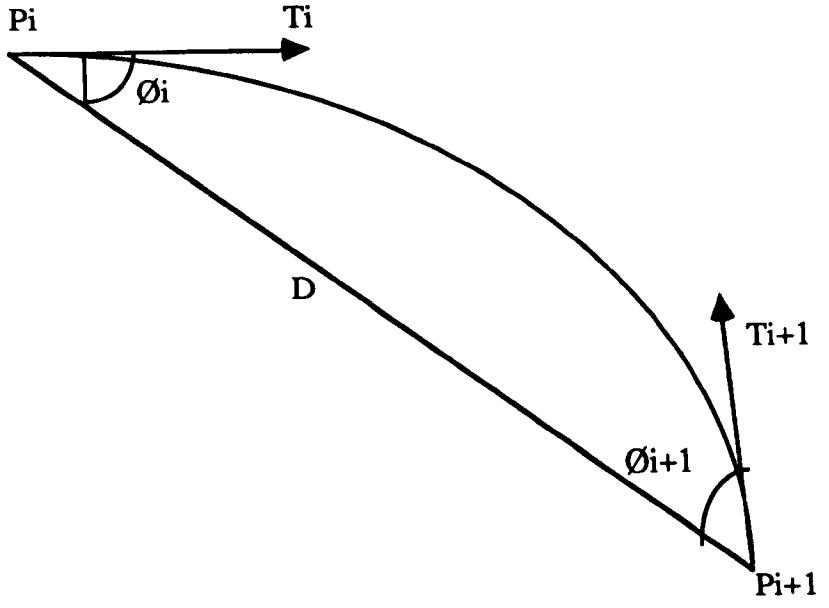


Fig.7.6 Asymmetric spline where in general $\emptyset_i \neq \emptyset_{i+1}$

So formulae (7.6) were adopted for defining the magnitudes of

$\mathbf{p_i'(0)}$ and $\mathbf{p_{i+1}'(1)}$

Results

Figures 7.7, 7.8, 7.9 and 7.10 show the approximation of handgenerated material by the linear splines, and cubic bezier representations. Each figure has four views. The bottom left is the original curve; the top left is the approximation; the top right is the approximation with relevant selected points superimposed and finally the bottom right is the approximation and original curve overlayed.

Before discussing the results let us remind ourselves the pictorial performance of the linear spline approximation. In what follows "L" refers to straight line approximation.

To illustrate again the linear spline approximation we present Fig.7.7. Fig.7.7.La is the original trajectory of the pen. Fig.7.7.Lb is the approximation. Fig.7.7.Lc is the approximation with the selected points

identified, and Fig.7.7.Ld is the original and the approximated traces superimposed.

Fig.7.7.Lb and Fig.7.7.Ld clearly illustrate the problems associated with only straight line approximation: If we look at the smooth parts of the pen trace, we can see that the smooth transition between approximated segments, is lost. Although the point continuity is ensured, the gradient continuity is lost, therefore kinks are easily detected by naked eye, this problem becomes accentuated when the picture is magnified (see previous chapter).

Further illustrations of straight line approximations are portrayed in figures 7.8, 7.9, 7.10 suffixed by L to indicate that we are looking at traces approximated by straight lines.

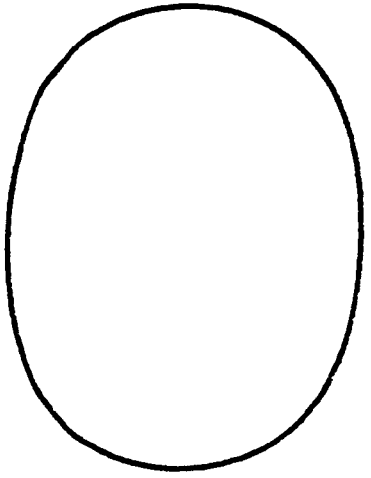
Now, we look at a few examples approximation by the representations and pass some qualitative evaluations.

A great deal of hand generated material was used to test our analytical representations, but for the purpose of the remainder of this chapter, the approximations will be described, with reference to original traces, Trace A, Trace B, Trace C and D respected pictured in Fig.7.6.a, Fig.7.6.b, Fig.7.6.c and Fig.7.6.d.

Using the cubic bezier polynomial for reconstruction, we obtain the pictures in Fig.7.7.CBa and Fig.7.7.CBd which respectively illustrate the bezier curve (Bezier curve in yellow and original trace in black) and bezier curve and original curve superimposed together for visual evaluation.

Further illustrations are shown in ,Fig.8.CBa, Fig.7.8.CBb, Fig.7.8.CBc and Fig.7.8.CBd which respectively show the original pen trace, the selected relevant points, the generated Bezier curve and Bezier polygon, the Bezier curve and the original curve superimposed.

Figures 7.9.CBa, 7.9.CBb, 7.9.CBc, 7.9CBd; 7.10.CBa, 7.10.CBb, 7.10.CBc give further illustrations.



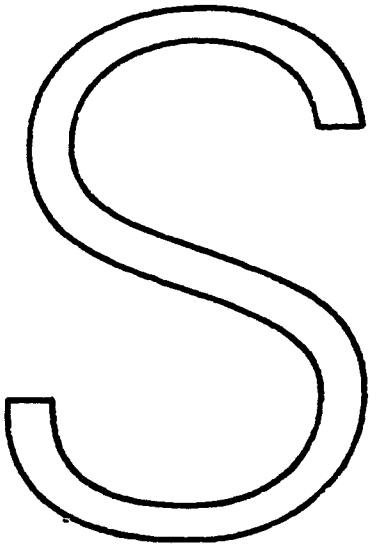
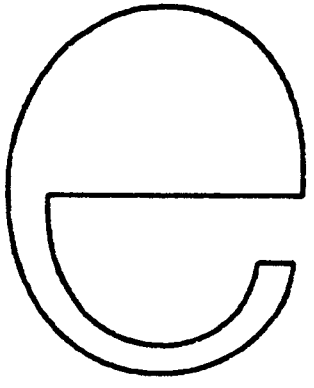
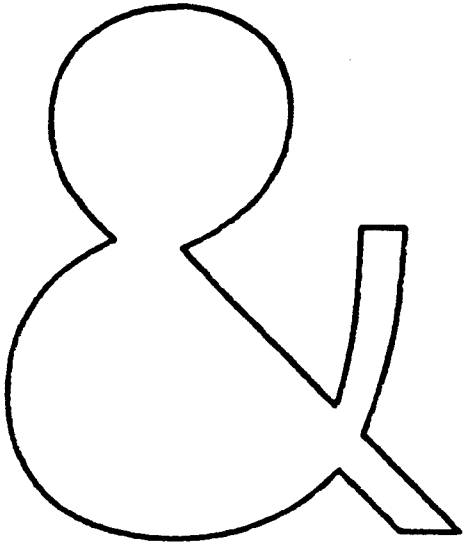
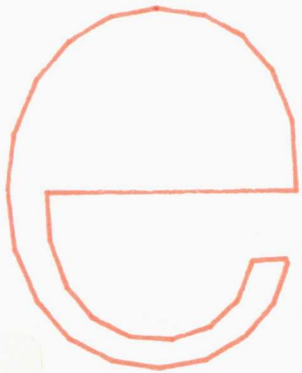


Fig.7.6
Clockwise, from the top left to the bottom left, figures 7.6.a , 7.6.b, 7.6.c 7.6.d are the original traces used for illustrative purposes.







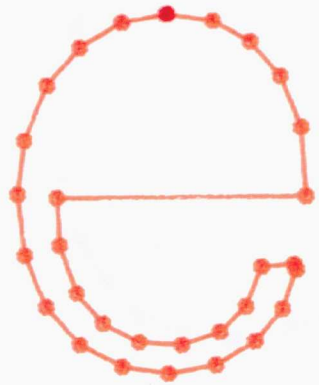
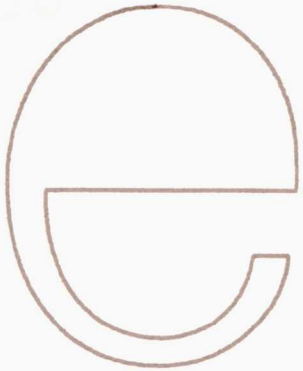
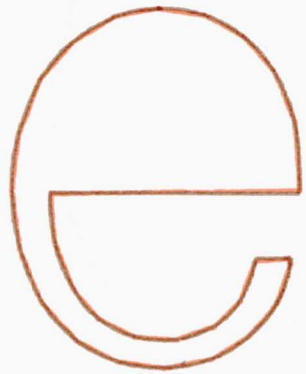
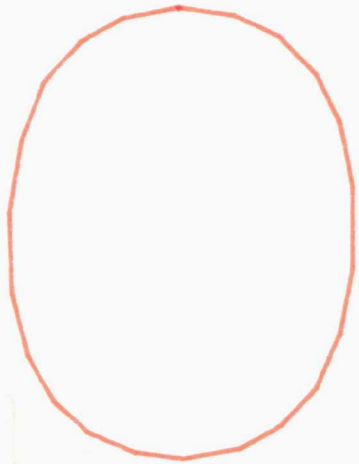
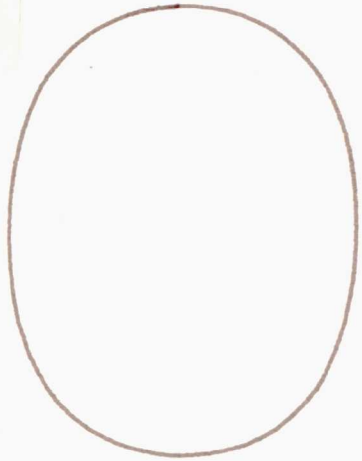


Fig.7.7
Clockwise from the bottom left to the bottom right, we have:
Fig.7.7.La Original trace.
Fig.7.7.Lb Straight line approximation
Fig.7.7.Lc Straight line approximation with relevant points.
Fig.7.7.Ld Approximated and original traces are overlayed









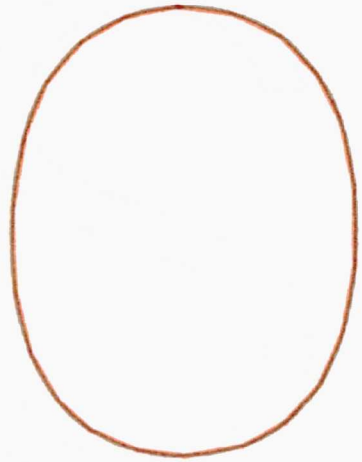
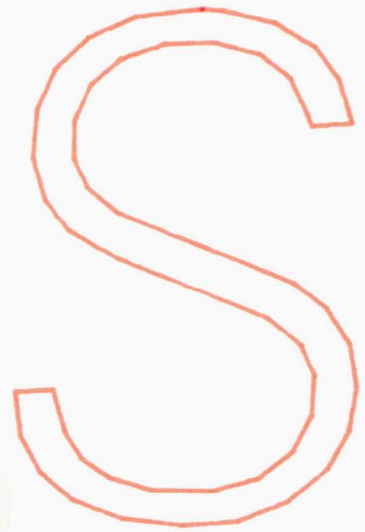
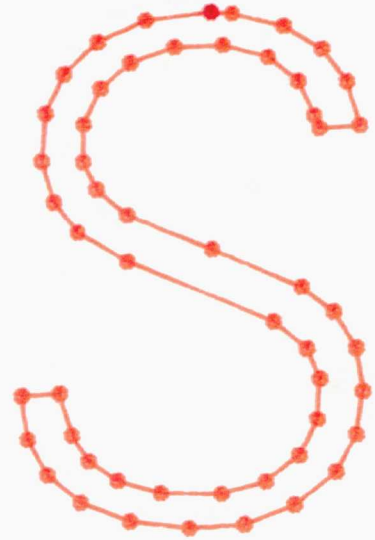
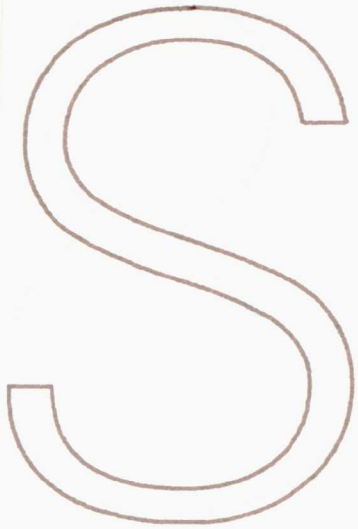


Fig.7.8
Clockwise from the bottom left to the bottom right, we have:
Fig.7.8.La Original trace.
Fig.7.8.Lb Straight line approximation
Fig.7.8.Lc Straight line approximation with relevant points.
Fig.7.8.Ld Approximated and original traces are overlayed







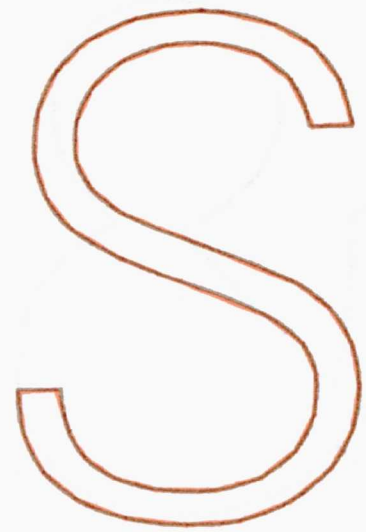
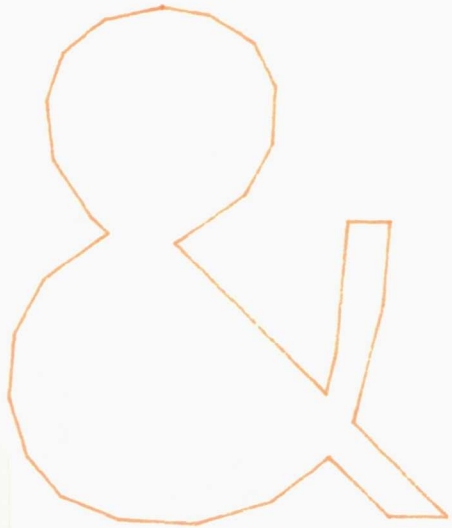
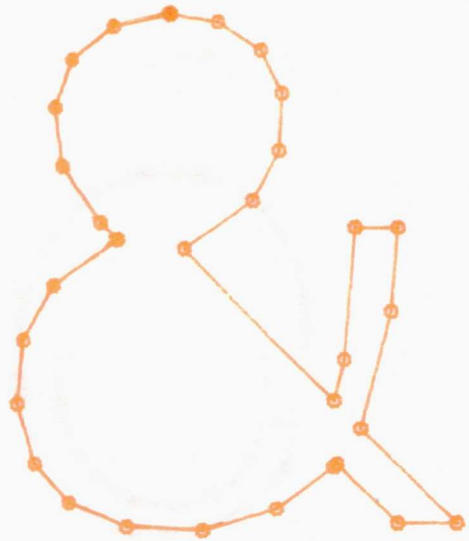
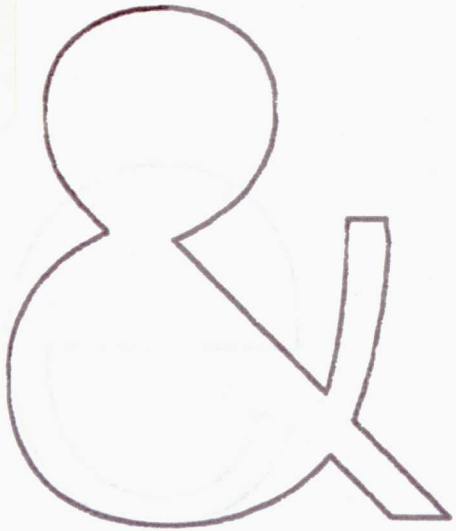


Fig.7.9
Clockwise from the bottom left to the bottom right, we have:
Fig.7.9.La Original trace.
Fig.7.9.Lb Straight line approximation
Fig.7.9.Lc Straight line approximation with relevant points.
Fig.7.9.Ld Approximated and original traces are overlayed







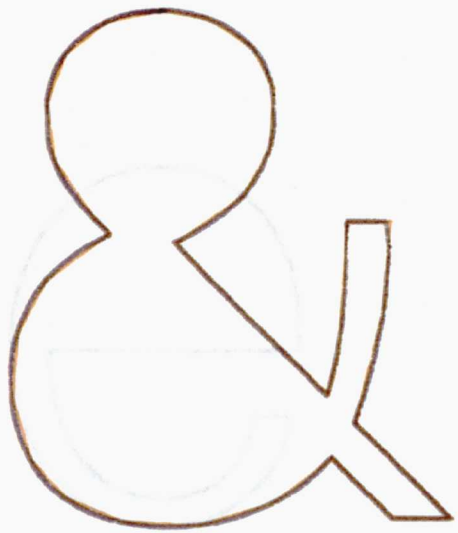
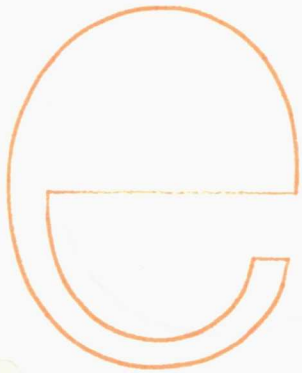
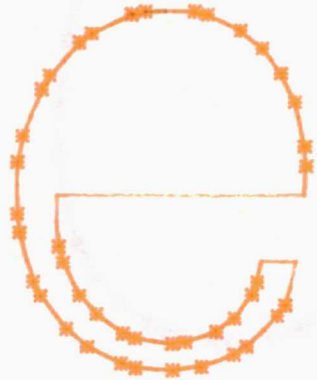
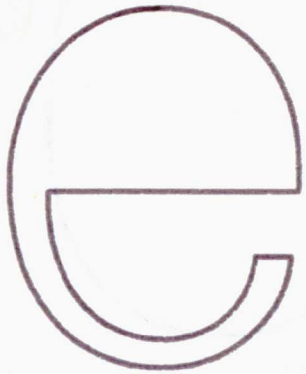


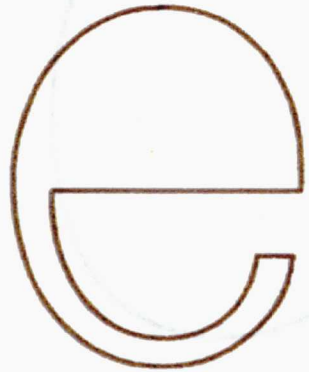
Fig.7.10
Clockwise from the bottom left to the bottom right, we have:
Fig.7.10.La Original trace.
Fig.7.10.Lb Straight line approximation
Fig.7.10.Lc Straight line approximation with relevant points.
Fig.7.10.Ld Approximated and original traces are overlayed

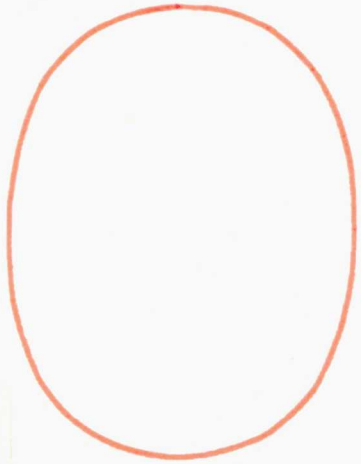




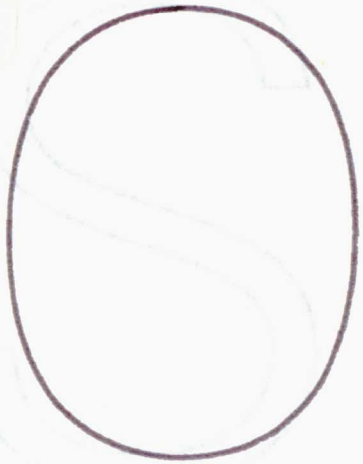
Clockwise from the bottom left to the bottom right, we have:
Fig.7.7.CBa : original trace.
Fig.7.7.CBb Bezier cubic approximation
Fig.7.7.CBc Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.7.CBd Bezier cubic approximated and original traces overlayed.

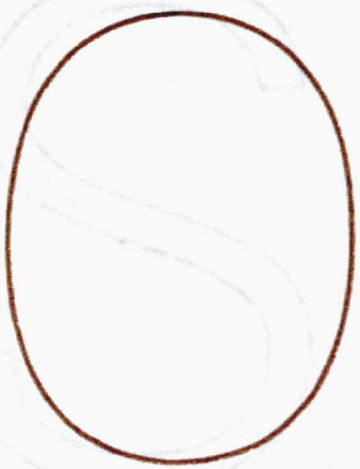


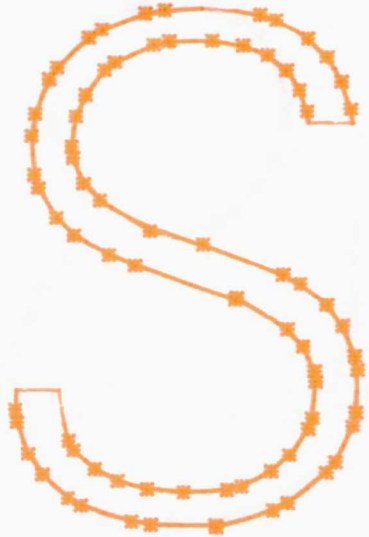




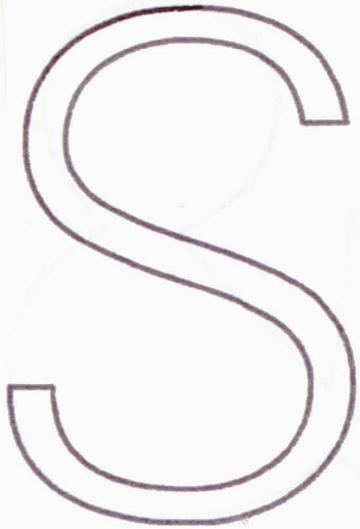
Clockwise from the bottom left to the bottom right, we have:
Fig.7.8.CBa : original trace.
Fig.7.8.CBb Bezier cubic approximation
Fig.7.8.CBc Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.8.CBd Bezier cubic approximated and original traces overlayed.

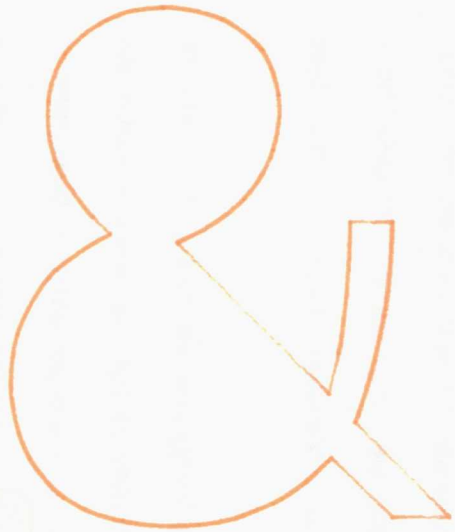


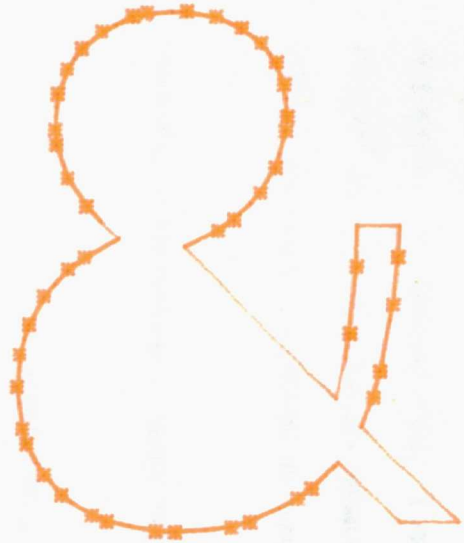




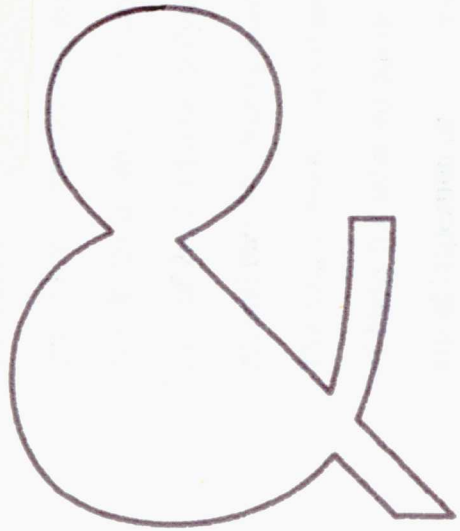
Clockwise from the bottom left to the bottom right, we have:
Fig.7.9.CBa : original trace.
Fig.7.9.CBb Bezier cubic approximation
Fig.7.9.CBc Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.9.CBd Bezier cubic approximated and original traces overlayed.

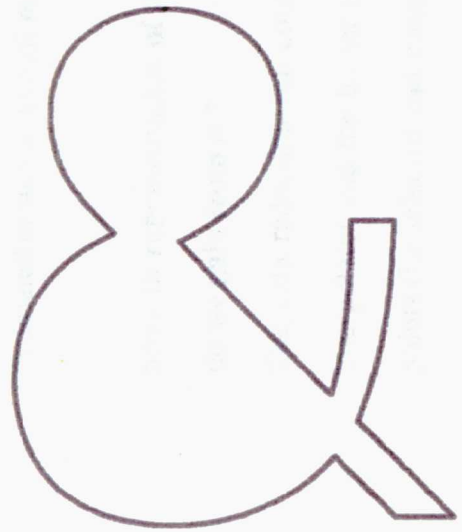






Clockwise from the bottom left to the bottom right, we have:
Fig.7.10.CBa : original trace.
Fig.7.10.CBb Bezier cubic approximation
Fig.7.10.CBc Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.10.CBd Bezier cubic approximated and original traces overlayed.





To define each piece of a pen trace, we note that the bezier cubic representations needs more points than straight line fits, this is no surprise because 4 points are necessary to specify a cubic, whereas only two points are needed for a straight line. In term of storage requirements, if the number of selected relevant points in straight line representation is m, then 16m bytes of storage are needed in general, because 2 coefficients must be stored for each of 2m first order polynomials, and each coefficient requires 4 bytes since in general, they are floating point values. As to the cubic bezier approximation, 32m bytes of storage are needed because four coefficients must be stored for each of 2m third order polynomials. Each coefficient requires 4 bytes since, in general they are floating point values.

The bezier approximation and the linear approximation have the same number of knots (i.e selected points) and represent the traces quite accurately, but a visual inspection of various examples shows that the bezier format has a slight edge. In the linear sections of curves all representations are reasonably accurate and efficient.

One would expect the bezier approximation to cause difficulties in the vicinity of the corners, but this does not happen because the way the gradients are estimated ensure a faithful reproduction of the corners

Smooth representation of the intended sections of the pen trace; but what do we gain from it ?

The data reduction (i.e compactness) has not changed because the straight interpolator and the bezier cubic interpolator use the same the interpolating points (i.e selected and transmitted points). The difference is to be found in the actual curve generation. Naturally it takes more time to generate a cubic polynomial than to generate a first order polynomial (straight line), how much more time and whether the effort is worth it, will be discussed in a later

paragraph.

When the interpolating points were selected, it is was generally assumed that the interpolator would be a linear spline. During the segmentation process in previous chapters, a straight line was fitted. Because a straight can fit two points, a quadratic can represent two points, a cubic can represent 4 points, and nth order polynomial can describe n+1 points, intuitively speaking, it is apparent that a straight line fitting approach is not necessarily efficient to segment a pen trace, if the interpolator is going to be a cubic. The obvious procedure is to fit a cubic instead, to help in selecting the relevant points. A coding scheme requiring the fitting of a cubic may be time consuming if every incoming point from the tablet is tested to see if the resulting approximation falls within a specified tolerance. Thus, there is a need to reduce the data points which go into the bezier cubic fitter. In this respect a new data reduction technique was developed for our purpose and published and can be found in CRAMPING85.

In the following paragraph, the data reduction will be described. The Bezier cubic is fitted sequentially to a consecutive points produced by our data reduction method, so instead of sending only selected points to the receiver, we send Bezier segments. Each Bezier segment is described by 4 consecutive points, two end points lying on the trajectory of the pen, and two control points lying on the tangent lines to the end points. At the receiver each Bezier formated segment is generated without the need for estimating gradient lines as above. So what we are actually saying is that, more work is done at the transmitter so that the receiver can only take care of the business of generating the curves which represent the trajectories of the pen.

7.3.2 Estimation of tangents during the pen trace segmentation.

Here the cubic bezier curve is fitted to a sequence points, so the pen trace is segmented by the process of fitting Bezier segments, thus a sequence of Bezier segments is sent to the receiver, which generates the curves using the data which specify each segment, and, of course as we have seen above, each segment is specified by four points; i.e 2 interpolating points and two control points. We adopt a scan along strategy, starting with three points, fit a bezier form cubic to the three points as follows :

We use the Pochop and Renner method presented above, to estimate the gradient lines at the first and the most recent incoming point. As the method requires 5 points, at the very first location of the pen, two more points are needed. Assuming that the first three points lie on a quadratic, using the simple process of extrapolation, we can obtain the first two points which are needed to estimate the gradient at the required point. The gradient at the most recent incoming point is estimated in the same way. When the process is started off, the gradient at the starting point is preserved until it is found that the most recent incoming data point could not join the previous points because the approximation lies outside the specified tolerance.

As with the linear spline representation, the approach is to reject the incoming point as long as the resulting approximation falls within a specified tolerance, that is

$|P - p(t)| \leq Tol$

where P is the most recent original sequence of points representing the trajectory of the pen, and p(t) is the cubic Bezier approximation of the original curve P. Tol is the specified tolerance; assuming that Tol is the maximum tolerable error, If k points have been captured so far, besides working out the gradient lines at the starting and at the kth point, k-2 interpolated points would have to be tested to see how far they are from from their original counterparts. Thus we can see that, this is an iterative process and could be very time

consuming if the number of the incoming points is large. To cut down the processing time, we need to reduce the number of points which go into the cubic Bezier fitter. Fig.7.10 portrays the work carried out in this paragraph.

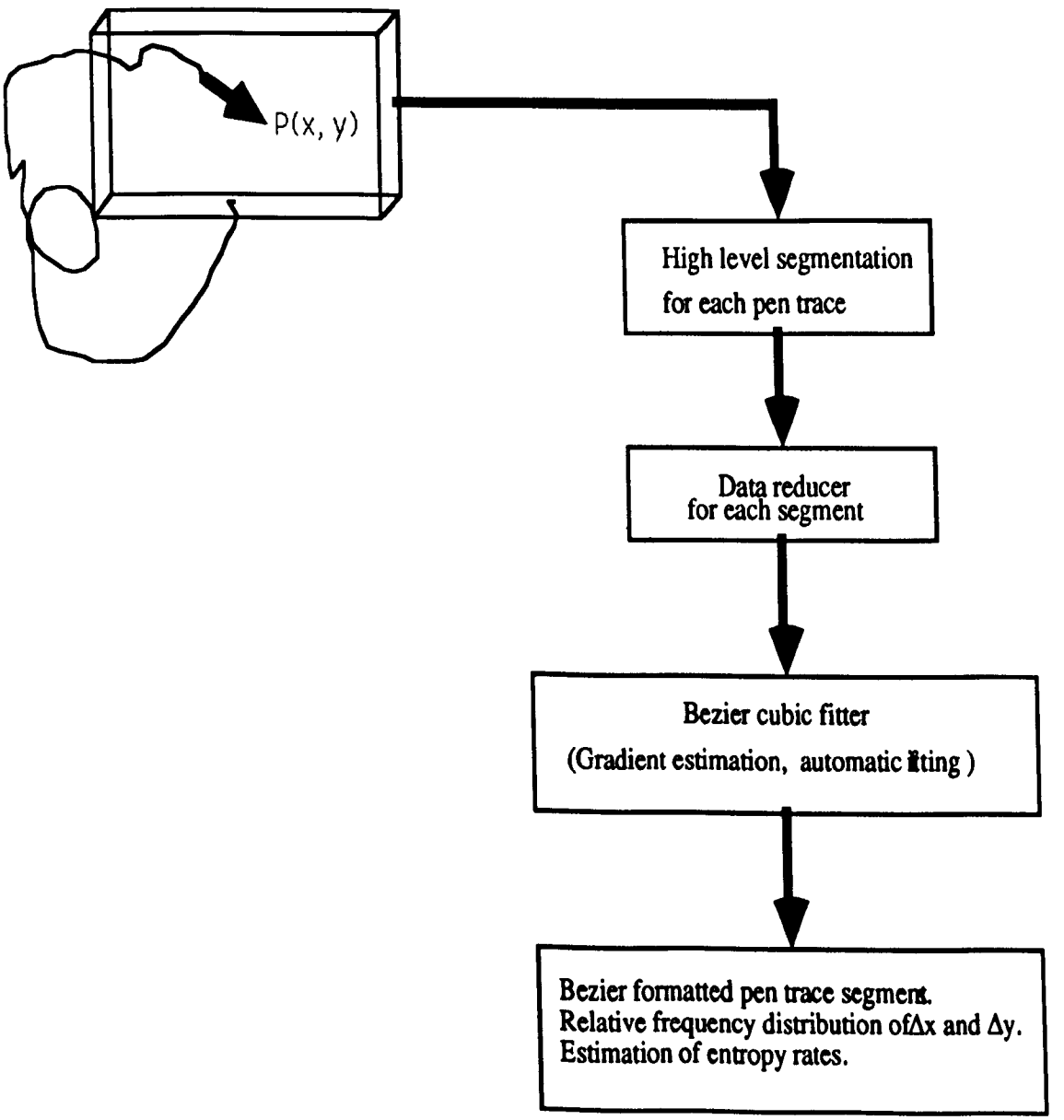


Fig.7.10 Processing stages for polynomial fitting

Looking at Fig.7.10, we need to describe four stages:

a. High level segmentation of each pen trace (see chapter 6).

b. The data reducing stage is applied to each segment produced by stage a.

c. The actual automatic cubic fitting process.

d. Differencing pocess is applied to the new data set formed by all the Bezier cubic formatted segments. The relative frequency distributions of Δx and Δy are used to estimate the entropy rates.

7.3.2.1 Data reduction

The work described here has been published in Computer aided design journal august 1985 issue, volume 17 number 6 (CRAMPIN85).

The object of the data reducing stage is to remove heavy work which would otherwise need to be done in the Bezier cubic fitting stage.

Our purpose is to describe an algorithm for the real time reduction of data while preserving the essential character of the handwriting which produced it. The problem is thus to devise a sensible real time process, between the information input (i.e Bit pad) and Bezier curve fitter, which will remove the redundant data while retaining the essential information. The simplest and most obvious, such process would be to ensure that each new piece of information is sent to the Bezier curve fitter, only if it represents a new pen position different from the previously pen position, to the bezier curve fitter. This simple process will be used as a "base line" in order to measure the efficiency of the more sophisticated method which we will describe shortly.

We shall assume that handwriting consists of a number of disjoint curves made up of smooth arcs separated by cusps, and the algorithm which we will describe falls naturally into two corresponding parts. The first part of the algorithm is a test for a cusp, and the second part involves a method of approximating smooth arcs by straight lines.

Linear approximation of smooth curves is, of curve a well established subject (PAVLIDIS74, PAVLIDIS82), but in this context there are two complications. Conventional method of linear interpolation will very often reduce the data at the expense of possibly losing the fine details of the curve; and in this context it is the fine detail which may convey the meaning of the handwriting. The second complication is that standard techniques are not particularly well suited to reducing data which must be processed and transmitted continuously.
The fundamental concept behind our data reduction method is the intuitively reasonable notion that, to interpolate a curve efficiently, one should place few points where the radius is large (i.e the curve is almost straight), and many points when it bends rapidly (i.e the radius of curvature is small). This the guiding principle behind this part of our algorithm, which caters for smooth arcs.

7.3.2.1.a Linear approximation of smooth curves

We suppose that $\{P_n\}$, with coordinates (X_n , Y_n), is the sequence of points which will be sent to the bezier curve fitter, and that we have just sent the element P_N. We then suppose that $\{P_k\}$ is the subsequent sequence of points

produced by the Bit pad, lying on a smooth arc γ. Our immediate objective is then to select an element P_{N+1} from $\{Pk\}$ to be the next transmitted point to the bezier curve fitter. We do this as follows:

Our algorithm compares the distance from P_N to a subsequent point Pk on γ,

with the minimum value , μ, of the radius of curvature ρ of γ between these points. If the distance $|P_NP_k|$ exceeds some function of μ, then Pk becomes the next transmitted point P_{N+1}.

It transpires that the appropriate function of μ is a multiple of its square root. Thus the first requirement of our algorithm is that it should estimate the radius of curvature ρ on γ and in order to achieve this, we select a subset $\{q_t\}$ from $\{P_k\}$ with the properties that

$$|q_t - q_{t-1}| \geq d \qquad (7.7)$$

$$|q_{t+1} - q_t| \geq d$$

and $P_1 = q_1 = P_N$, for some constant $d > 0$.

We then use the radius r(t) of the circle passing through the three points q_{t-1} , q_t , q_{t+1} as an estimate for the radius of curvature of ß at the point q_t. It can be shown (see Appendixchap7) that

$$r(t) = |q_{t-1} - q_t|\,|q_t - q_{t+1}|\,|q_{t-1} - q_{t+1}| / 2\Delta \qquad (7.8)$$

Δ is defined in the appendix as twice the area of the triangle $(q_{t-1}\, q_t\, q_{t+1})$

Equation (7.8) is our approximation for $\rho(t)$. At this stage of our algorithm, we need make no special provision for the case $\Delta = 0$ since this is previously eliminated by our method of detecting straight line and cusps.

Now let

$\mu = \min r(t)$, from P_N to q_t and we then use the criterion

$$P_{N+1} = q_t \text{ if } |q_t - P_N| \geq K\sqrt{\mu} \qquad (7.9)$$

where K is some suitably chosen constant, in order to select the next transmitted point P_{N+1}.

The choice of the constant d

The value of the constant d is a compromise between two conflicting requirements. We require d to be sufficiently large for equation (7.8) to give a reasonable estimate for $\rho(t)$; but we also require d to be small in order to retain the possibility of sending points separated by one unit of the input device, to the Bezier curve fitter. Actually there is a further reason for choosing d to be small as we shall see when discussing cusps.

We choose d = 1, so that it is possible for three points qt-1 qt qt+1 to be consecutively distant from each other by one unit corresponding to the resolution of the input device.

We have thus produced a method of selecting a sequence $\{P_N\}$ of points to be transmitted to the cubic bezier fitter, and we know that between each pair of points, P_N and P_{N+1} say, we are trying to reproduce a curve γ with radius of curvature $\rho \geq \mu$. Our task now is to estimate the error introduced when we replace γ by the straight line P_NP_{N+1}.

Ideally we should like the output from our process to be identical to that of the simple process mentioned at the beginning of this section.

Error estimates

Before we deal with the error estimates, let us recall the following results of differential geometry, which help to understand the analysis:

Given a planar function function y = f(x), if S is the length of the curve, the radius of curvature is given by $\rho = dS/d\phi$ and

$dx/d\phi = \cos\phi$; $dy/d\phi = \sin\phi$;

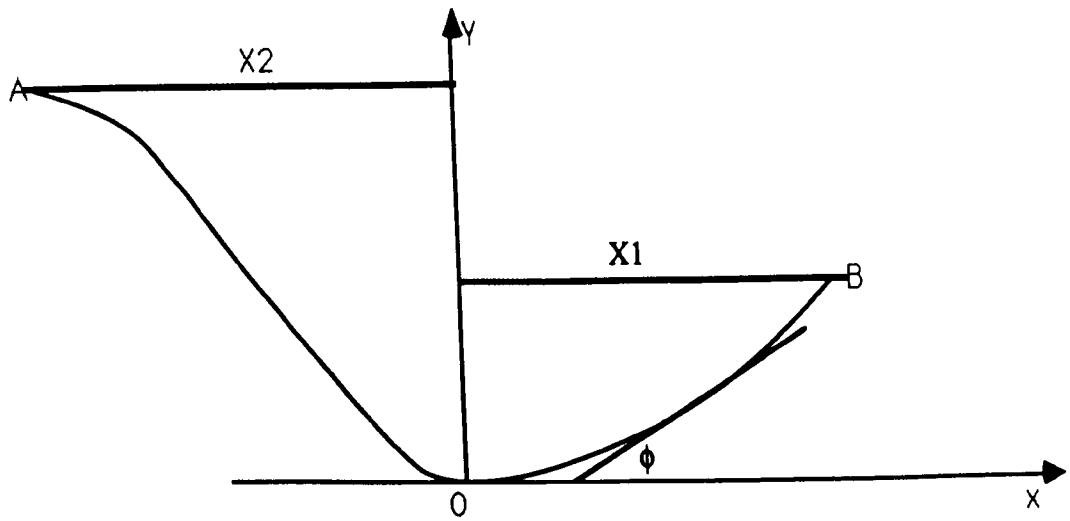
ϕ is the gradient at a given point (x, y) of the curve.

Let Γ_{AB} be a smooth curve of fixed length S_o, with ends A and B, such that the radius of curvature $\rho \geq \mu > 0$, for some constant μ, on Γ_{AB} ;

and $So < (\mu\pi)/2$ then

$|AB| \geq 2\mu\sin(So/2\mu)$

Proof



X1 and X2 are respectively the abscissas of points B and A, with common ordinate $Y = \varepsilon$.

Let O be any point on the arc Γ_{AB} and choose coordinates as shown.

IF S denotes the length on Γ_{AB}, with O as origin, then $\rho = dS / d\phi$, so that

$$\phi(t) - \phi(0) = \int_0^t (d\phi/dS)\, dS = \int_0^t 1/\rho\, dS$$

where $t \leq S \leq S_1$ and S_1 = arc length Γ_{OB}

But $\phi(0) = 0$ and $1/\rho \leq 1/\mu$ on Γ_{OB}

so that $\phi(t) \leq \int_0^t 1/\rho \, dS$

and therefore $\phi(t) \leq t/\mu \leq So/\mu < \pi/2$ **(7.10)**

It follows that $\cos(\phi(t)) \geq \cos(t/\mu)$ for $0 \leq t \leq S_1$.

This estimate can now be used to obtain an inequality for X_1 as follows

$$X1 = \int_0^{S1} \cos(\phi(s)) \, ds$$

$$\geq \int_0^{S1} \cos(s/\mu) \, ds$$

$$= \mu \sin(S1/\mu) \qquad (7.11)$$

Similarly we can see that

$X2 \geq \mu \sin(S2/\mu)$

where S2 = arc lengh OA

thus $|AB| \geq X1 + X2 = \mu (\sin(S1/\mu) + \sin(S2/\mu))$

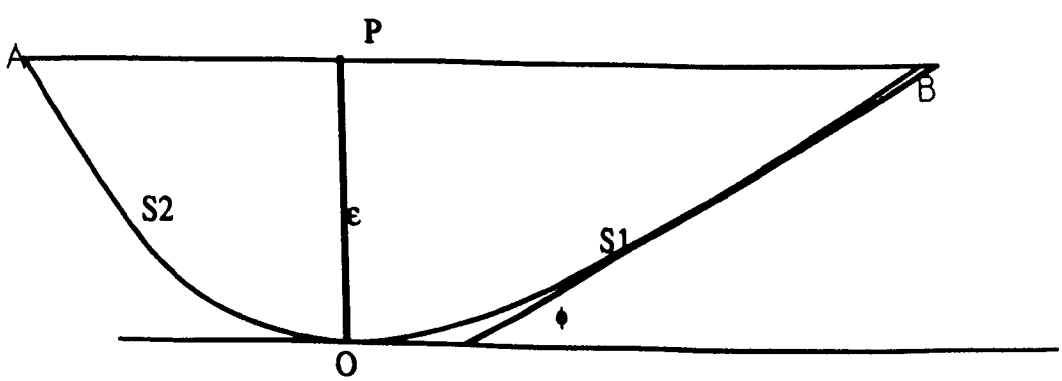
$$= 2k\sin(So/2\mu).\cos(S1 - S2)/2\mu$$

We may choose the point O on Γ_{AB} so that S1 = S2 = So /2, in which case we have $|AB| \geq 2\mu\sin(So/2\mu)$ as required .

Let ε be the maximum perpendicular distance from Γ_{AB} then

$\varepsilon \leq \mu (1 - \sqrt{(1 - (|AB| / 2\mu)^2)})$

proof



Let O be the point on Γ_{AB} such that the perpendicular distance is greatest , then

$\phi(0) = 0$ and using equation (7.10)

$$Y = \int_0^{S1} \sin(\phi(s))\, ds$$

$$\leq \int_0^{S1} \sin(s/\mu))\, ds$$

$$= \mu\,(1 - \cos S1/\mu\,)$$

It follows that $S1 \geq \cos^{-1}(1 - \varepsilon/\mu)$

and similarly $S2 \geq \cos^{-1}(1 - \varepsilon/\mu)$

Thus $So = S1 + S2 \geq 2\cos^{-1}(1 - \varepsilon/\mu)$

so that $\varepsilon \leq \mu(1 - \cos(So/2\mu)\,)$

$$= \mu(1 - \sqrt{(1 - \sin^2(So/2\mu)\,)}\,)$$

$$\leq \mu(\,1 - \sqrt{(1 - (\,|AB| / 2\mu\,)^2)}\,) \qquad (7.12)$$

This inequality is best possible since for a circle

$$\varepsilon \leq \mu\,(\,1 - \sqrt{(1 - (\,|AB| / 2\mu\,)^2)}\,)$$

If we now assume that |AB| is small compared to 2k then

$$\varepsilon \leq \mu\,(\,1 - (1 - (\,|AB| / 2\mu\,)^2)^{1/2}\,)$$

$$\approx k\,(|AB|^2\,)/\,8\mu^2 \quad + O(|AB|2\mu)^4$$

$$= |AB|^2 / 8\mu \quad + O(|AB|/2\mu)^4$$

Take inequality (7.12) i.e

$$\varepsilon \leq \mu\,(\,1 - \sqrt{(1 - (\,|AB| / 2\mu\,)^2)}\,)$$

we can see that

$1 - \varepsilon/\mu \geq \sqrt{(1 - (|AB| / 2\mu)^2)}$

so that $(1 - \varepsilon/\mu)^2 \geq 1 - (|AB|^2 / 4\mu^2)$

Hence $|AB|^2 \geq 8\mu\varepsilon - 4\varepsilon^2$

So if we want to achieve an error of less than one unit of the input device, then we need $|AB| \geq \sqrt{(8\mu - 4)}$

In equation (7.9) we now choose qt to be the first point for which $|qt - PN| \geq \sqrt{(8\mu)}$ choosing K to be $\sqrt{8}$. Assuming that the pen is not moving very rapidly across the digitising tablet, we will have

$$|qt - PN| = \sqrt{(8\mu)} \quad (7.13)$$

so that the linear interpolation will produce an error of less than one pixel on each section of γ. If, for example, we were prepared to accept an error of at most two pixels then we would choose $K = \sqrt{16} = 4$

Thus if we adopt (7.13) as part of our algorithm we will have

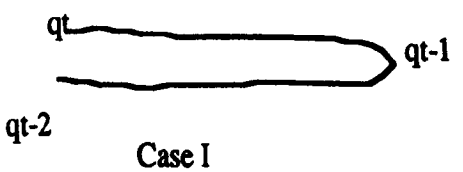
$$|PN - PN+1| = \sqrt{(8\mu)} \quad (7.14)$$

and this will ensure an error of less than one pixel.

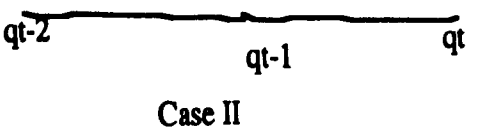
7.3.2.1.b Straight lines and cusps

In our calculation of r(t) in equation (7.8) we have assumed that $\Delta \neq 0$, since this case is eliminated as follows.

There are actually two cases to consider .



Case I



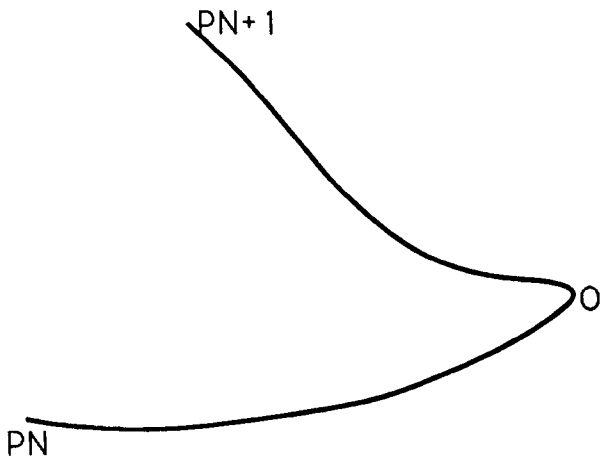
Case II

For case I , if $\Delta = 0$ and dot product $q_{t-1}q_t \cdot q_tq_{t+1} < 0$ then we set r(t) = 0, and restart the algorithm.

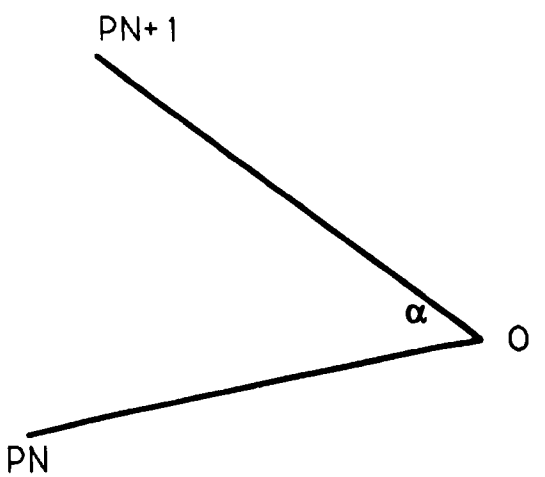
For case II, if $\Delta = 0$ and $q_{t-1}q_t \cdot q_tq_{t+1} > 0$ then we set $r(t) = \infty$.

Our previous algorithm, although designed for smooth curves, is quite capable of dealing with cusps which are not too sharp; but for sharp cups we will need to introduce a further refinement.

Suppose that there is a cusp at the point O and that P_N and P_{N+1} are successive interpolation points.

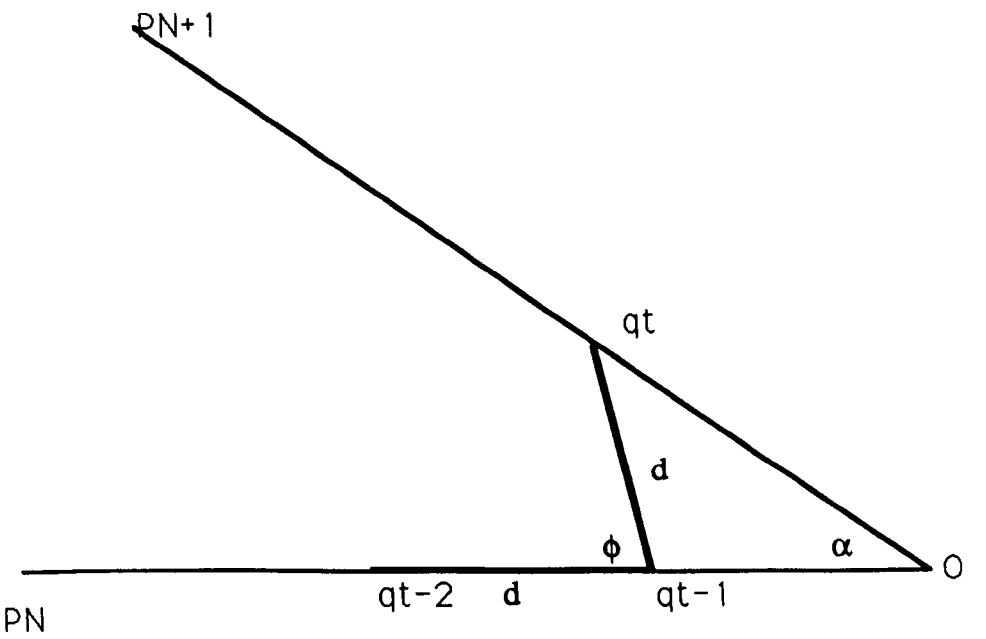


Our algorithm for smooth curves ensures that the straight line P_NO differs by less than one pixel from the curve γP_NO, so that we are justified in approximating the cusp by the intersection of the two straight lines, thus

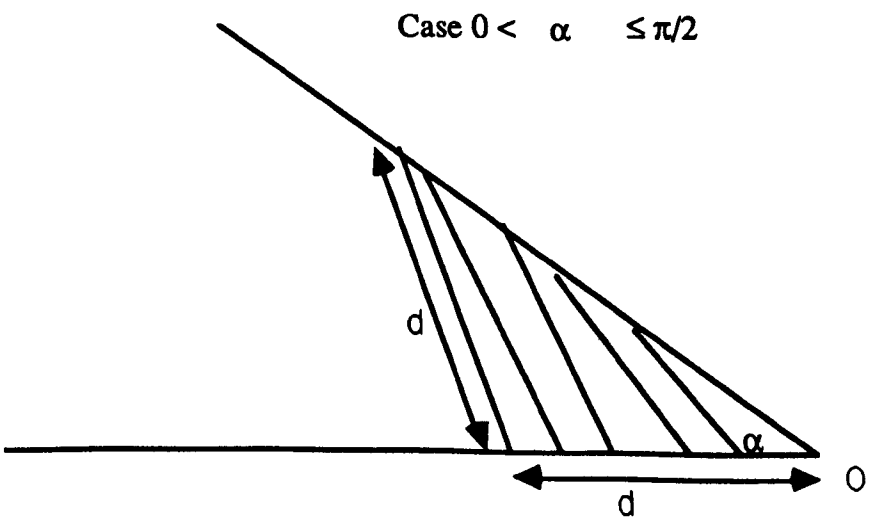


intersecting at an angle α say.

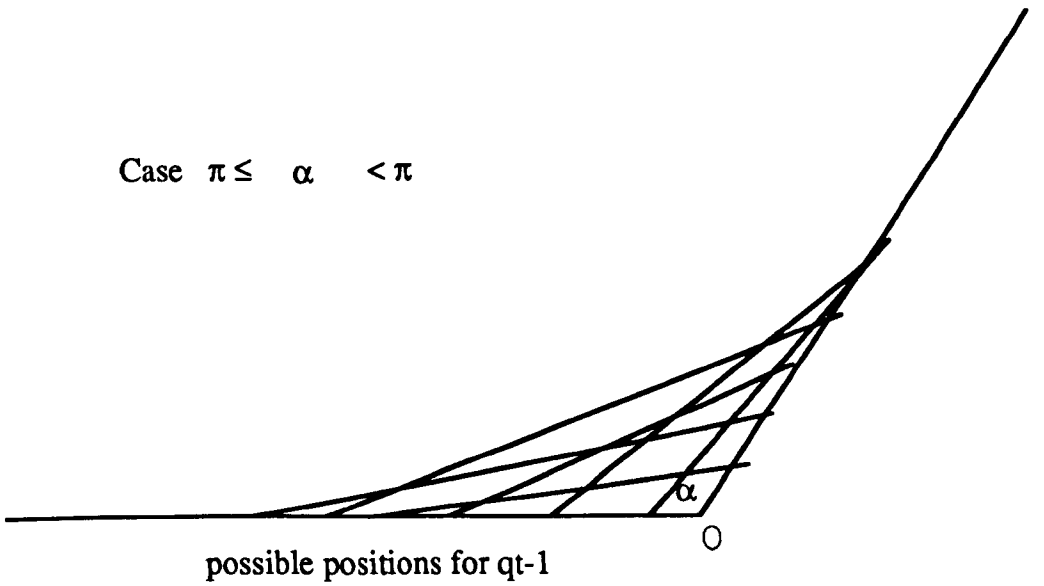
Consider now our process for calculating r(t) when $0 < \alpha < \pi/2$



The estimate which we obtain for $\mu = \min r(t)$ on $\gamma P_N P_{N+1}$ will depend on the actual positions of the points {qt} in relation to the point O but the smallest calculated value will occur when the angle ϕ is least, and equal to ϕ_O, say



Case $\pi \leq \alpha < \pi$

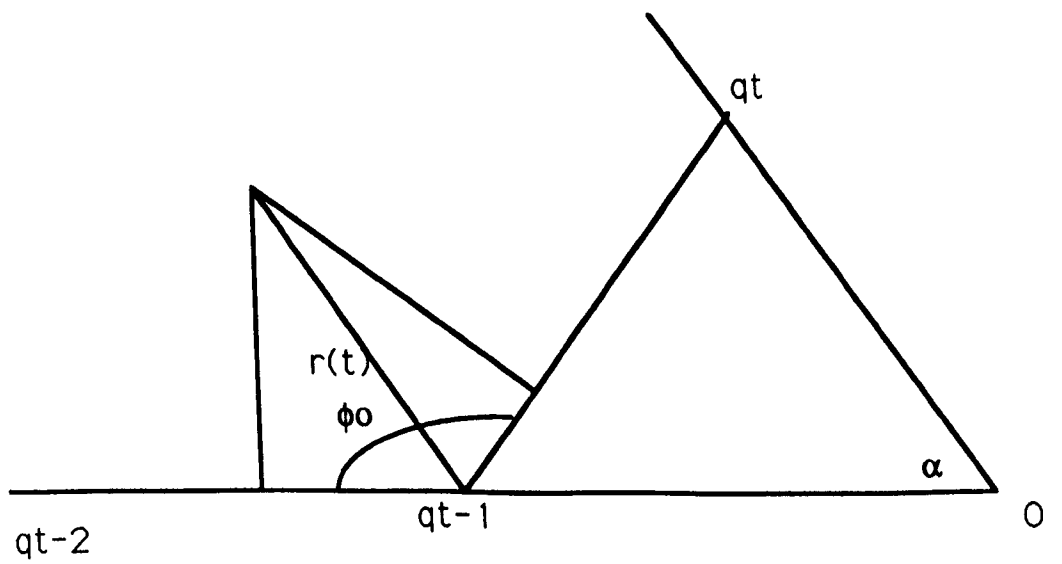


possible positions for qt-1

For a particular polygonal arc $\{q_t\}$ from P_N to P_{N+1}, for a fixed angle α;

ϕ_0 will be the least value of the angle ϕ between successive straight lines. Our objective is to calculate the greatest possible value of ϕ for a fixed angle α. We recall that qt is the first element of the sequence {Pk} for which $|q_t - q_{t-1}| = d$, and then consider two cases:

Case 1: $\pi/3 \leq \alpha < \pi$



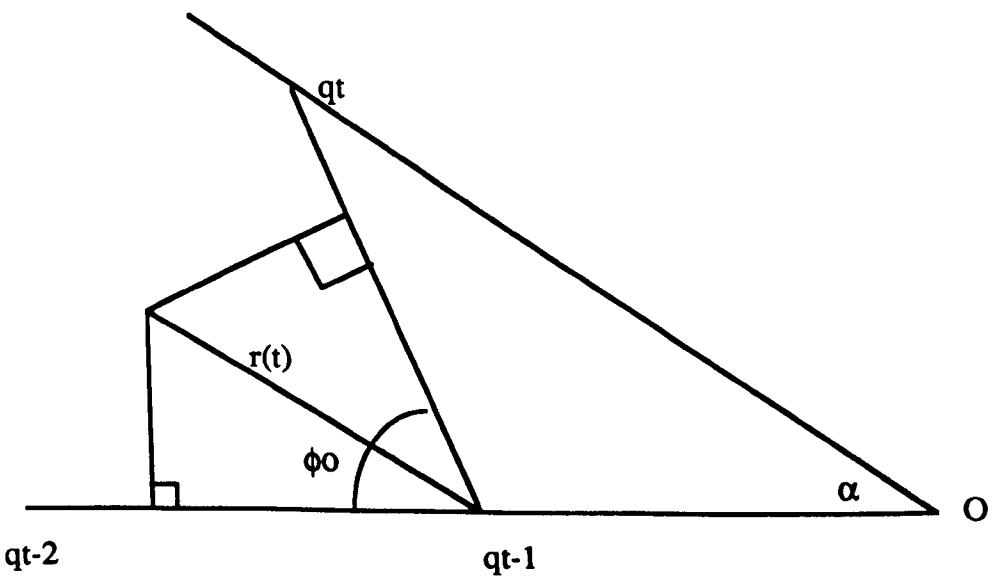
In this case the greatest value of ϕ occurs when the triangle $q_{t-1}Oq_t$ is isoceles

(with $|q_{t-1}O| = |Oq_t|$), in which case we have $\phi_0 = (\pi + \alpha)/2$,

and $\mathbf{r(t) = d / (2\cos((\pi + \alpha)/4))}$ **(7.15)**

It follows that $\mu \leq \mathbf{d / (2\cos((\pi + \alpha)/4))}$ **(7.16)**

Case 2: $0 < \alpha < \pi/3$



We notice that $|q_{t-1}O| \leq d$;

and that the limiting case occurs when $|q_{t-1}O| = d$, in which case $\phi_0 = 2\alpha$.

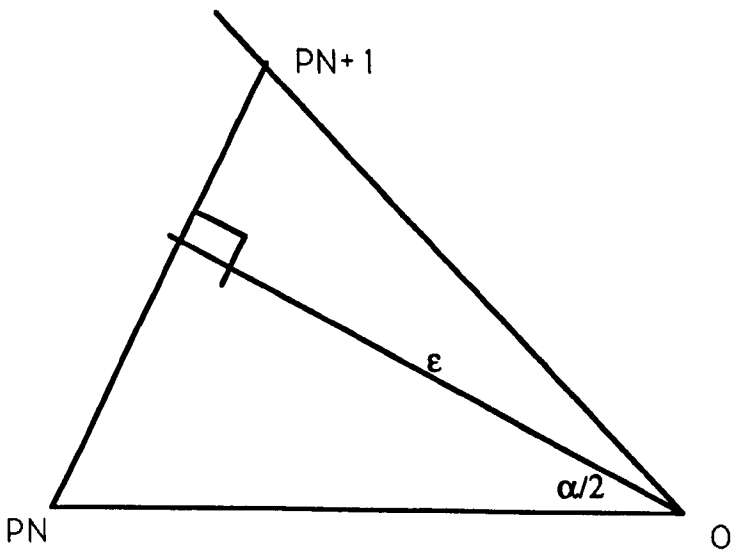
It follows that, when ϕ_0 is less than its greatest value

$r(t) \leq d/(2\cos\alpha)$

so that $\mu \leq \mathbf{d/(2\cos\alpha)}$ **(7.17)**

Notice particularly that inequalities (7.15) and (7.17) are best possible estimates, although in both cases we might expect to get considerably smaller values of μ depending on the position of q_{t-1} relative to O.

Error estimates.



The error ε, in the sense of the perpendicular distance from O to P_NP_{N+1}, is greatest when the triangle P_NOP_{N+1} is isoceles, in which case, using (7.14), we have

$$\varepsilon = (\sqrt{8}\mu)/ (2\tan\alpha/2) \qquad (7.18)$$

We can examine the following conditions :

Condition a: If $\pi/3 \leq \alpha < \pi$ we have from (7.16) and (7.18)

$$\varepsilon = ((\sqrt{8}) / (2\tan\alpha/2)) \ (\sqrt{d}/(\sqrt{2}\sqrt{\cos((\pi + \alpha)/4)}))$$

$$\varepsilon = ((\sqrt{8}) / (2\tan\alpha/2)) \ (\sqrt{d}/(\sqrt{2}\sqrt{\cos((\pi + \alpha)/4)}))$$

$$= \sqrt{d} / (\tan\alpha/2 \sqrt{\cos ((\pi + \alpha)4)}) \qquad (7.19)$$

which is a decreasing function of α in the range $\pi/3 \leq \alpha < \pi$

It follows that putting $\alpha = \pi/3$

$$\varepsilon \leq \sqrt{6}d$$

If we choose d = 1 then $\varepsilon < \sqrt{6}$, so that we have an error of at most 2 units of the input device. In fact we would expect that even this error is unlikely, depending on the position of q_{t-1} to O.

Thus for angles in the range $\pi/3 \leq \alpha < \pi$ we need make no special provision in our algorithm.

Condition b: If $0 < \alpha \leq \pi/3$ we have from (7.16) and (7.18)

$$\varepsilon \leq ((\sqrt{8})/ (2\tan\alpha/2)) (\sqrt{d}/(\sqrt{(2\cos\alpha)})$$

$$= \sqrt{d} / (\tan\alpha/2 \sqrt{\cos\alpha}) \qquad (7.20)$$

and this estimate is the best possible. Thus for small angles α our algorithm may well give rise to unacceptable errors. We therefore adapt our algorithm in the following way. We examine each point $\{P_k\}$, as it arrives from the digitising tablet and compute the length $|P_N - P_k|$. If this length is greater than all previously calculated values then we set Q = pk. Thus Q is always the point on the curve γPNPN+1 furthest from PN. Before plotting the point PN+1, we test its distance from PN as follows

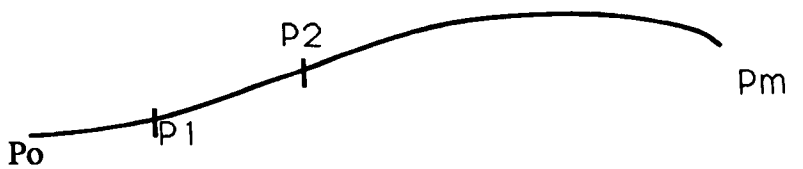
If $|P_{N+1} - P_N| \leq |Q - P_N|$ then $P_{N+2} = P_N$ and $P_N = Q$.

In the case of a cusp with $\alpha \leq \pi/3$, the point Q will be that apex of the cusp

7.3.2.1.c Efficiency of the algorithm

If we let M be the number of points which would be sent to the bezier curve fitter after the simple reduction process described in section 7.3.2.1, and let m be the number of points transmitted after our algorithm. Then the ratio $C = m/M$ is a measure of the efficiency of the method.

Let Γ be a smooth curve on which our method produces the captured points $\{P_N\}$ for N = 1, 2, ...m



$$\text{Then } \int_\Gamma 1/\sqrt{\rho}\, ds \approx \sum_{N=0}^{N=m-1} \int_{PN}^{PN+1} 1/\sqrt{\mu}\, ds$$

$$\approx \sum_{N=0}^{N=m-1} |P_{N+1} - P_N|\ 1/\sqrt{\mu}$$

$$\approx \sum_{N=0}^{N=m-1} 8 = 8m$$

$$\text{Also } \int_\Gamma ds \approx M$$

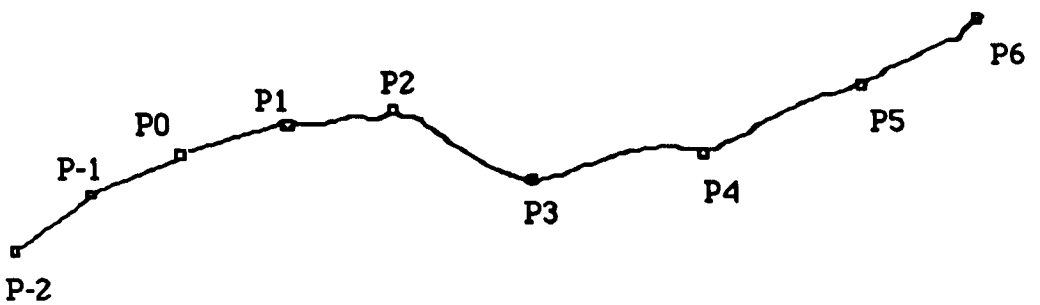
$$\text{so that } \left(1/8 \int 1/\sqrt{\rho}\, ds \right) / \int_\Gamma ds = m/M$$

In other words the mean value of $1 / (8\sqrt{\rho})$ is a measure of the reduction ratio of our method. For example, for a circle of radius of 16 units, we would expect a reduction ratio of 1/ 32, and for larger circles this ratio would be even smaller. Our algorithm is thus efficient for reduction of data while preserving the desired picture.

7.3.2.1 Automatic fitting of bezier cubic to a stream of consecutive pen positions.

The objective here is to fit a bezier cubic to points generated by the data reductor, without user interaction. As well, the generation should be in real time. This means that completion of a pen trace should not be necessary before fitting can begin, but a small time delay would be acceptable.

To start off the fitting process, the bezier curve fitter, waits to receive five points from the data reducer. With reference to the following illustration,



P-1 and P-2 are phantom points

the first trial starts from P_0 to P_2. Using Renner and Pochop's method, an estimate of the slope at P_0, requires two extra points (phantom points) P_{-2} , P_{-1}, which are found by extrapolation based on passing a quadratic polynomial through the first three points P_0 P_1 P_2. The slope at P_2 is estimated using the first five genuine pen positions P_0 to P_4 sent by the data reducer.

The first fitting trial is carried out by setting the magnitudes of the gradient, as explained previously. To determine the acceptability of the fitted curve, we must estimate the maximum error, between the original

(i.e smoothed curve from data reducer) and the analytical curve. There is not a one to one mapping between the original data points and the points generated from the mathematically fitted curve. So it is not obvious how to estimate the distances between the fitted curve and the original points. The strategy we use

depends on finding the minimum distance between the point and the analytical curve for each original pen position. This is done as follows :

The square of the distance between a given point x, y and any point (X(t), Y(t)) that lies on the curve is

$$(X(t) - x)^2 + (Y(t) - y)^2$$

This distance is minimum, if the point (x, y) lies on a normal to the curve (X(t) , Y(t)). To estimate this minimum, we differentiate and equate to zero, which yields

$$(X(t) - x)X'(t) + (Y(t) - y)Y'(t) = 0 \qquad (7.21)$$

where X'(t) and Y'(t) are the first derivatives with respect to t of parametric expressions X(t) and Y(t) with $0 \leq t \leq 1$.

Another way to look at it, is to find the value of t which ensures that a normal to the curve (X(t), Y(t)) passes through an original point (x, y). We know that the equation of the line with gradient m, passing through a point x, y is $Y - y = m(X - x)$, its corresponding normal is $Y - y = -1/m(X - x)$.

We can transport these expressions to parametric equations X(t), Y(t) by setting $m = (dY(t)/dt) / (dX(t)/dt)$, and substituting X(t) and Y(t) into the normal equation produce exactly the equation (7.21), which, when expanded, with due consideration to the cubic bezier expressions of (7.3), produces a fifth degree polynomial in t. We found Newton's method for solving a quintic equation was effective, and usually converged after at most four iterations.

All the distances between the curve and the original points were estimated, as just explained, and the larger distance was extracted for comparison with the tolerance. If the maximum distance is acceptable, we extend our trace segment by moving to the next point, P3, where we work out the gradient by taking into consideration points P1, P2, P3, P4, and P5 which is the most recent point

from the data reducer, and repeat the fitting. We should note that as long as the fitting is acceptable, we carry on in the same way, and the gradient at P0 is not altered. Should the fitting be unacceptable, adjustments are made on the magnitudes of the gradients, and the test for acceptability or rejection is carried out. This is an iterative process; so questions must be raised about the details of the method :

1. What are the starting values for gradients ?
2. Given a maximum error, how do we update the gradients ?
3. What are the stopping conditions for the algorithm ?
4. What is the stability and efficiency of the method ?

These questions are dealt with in this section.

Finding starting values for gradients is one of the easier problems. The slope estimate uses more of the information present in the original curve.
Renner and Pochop's method was adopted for this purpose.
How to update the gradients is a complex question. Two basically different approaches exist, although hybrid methods are possible. The first approach is to adjust the magnitude only, while the second is to recalculate the new gradients (magnitudes and directions upon each iteration).
Adjusting the magnitude is a far simpler and more stable approach. We rejected the adjustment of the gradient direction because it is possible to end up with a direction for the gradient that does not correspond at all to the local direction of the original curve. For this reason, only the magnitude adjusting fit is examined further. With reference to Fig.7.11 below,

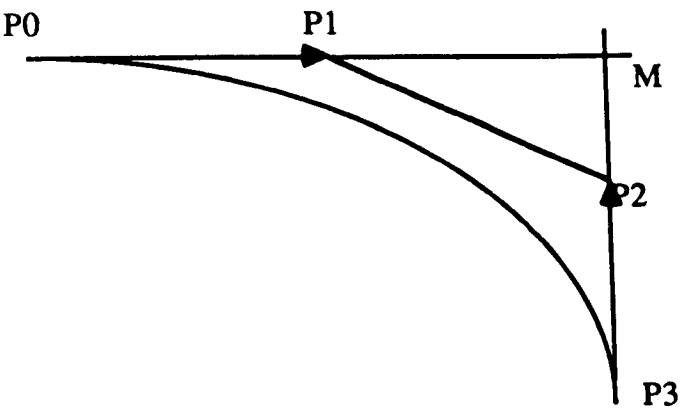


Fig.7.11 Bezier segment, tangent lines meet at M.

the magnitude adjustment is done by placing P1 and P2 on the fixed tangents at P0 and P3. We have two options:

1. The lengths of P0P1 and P2P3 can be adjusted simultaneously to give greater fullness to the curve.
2. We can adjust the lengths of P0P1 and P2P3, differentially in order to draw the curve nearer to one or other tangent.

As we are dealing with planar curves, the curvature is always towards the chord P0P3 if the points P1 and P2 lie within the segments P0M and MP3, where M is the intersection of P0P1 and P2P3.

It has been found that when P0P1 > P0M and P2P3 > MP3, the curve may have a loop. FORREST68, suggests that P0P1 and P2P3 should never exceed the cord P0P3, this prevents loops occurring when $0 \leq t \leq 1$. This implies that any loops occurring in curves obeying this rule will always lie outside the range $0 \leq t \leq 1$.

Gradient magnitudes are updated as follows :

When the gradient magnitudes are chosen, we calculate the distances of the original points to the mathematical curve. To the maximum distance are associated an original point and its corresponding counterpart on the mathematical curve. By comparing, the distances of the two points to the chord, P0P3, we are in the position to determine whether the fitted curve in that

region is above or below the original curve. The position of the point with respect to the mid point tells us which tangent magnitude should be adjusted. The sign of the error tells us which way the magnitude should be adjusted. If the error is positive, the gradient magnitude should be increased, and if the error is negative the gradient magnitude should be decreased. The sign of the error is found by subtracting the distance to the chord of the point derived from the analytical curve, from the distance to the chord of the original point. With reference to Fig.7.12, this means the sign of the differences of distances CA and BA.

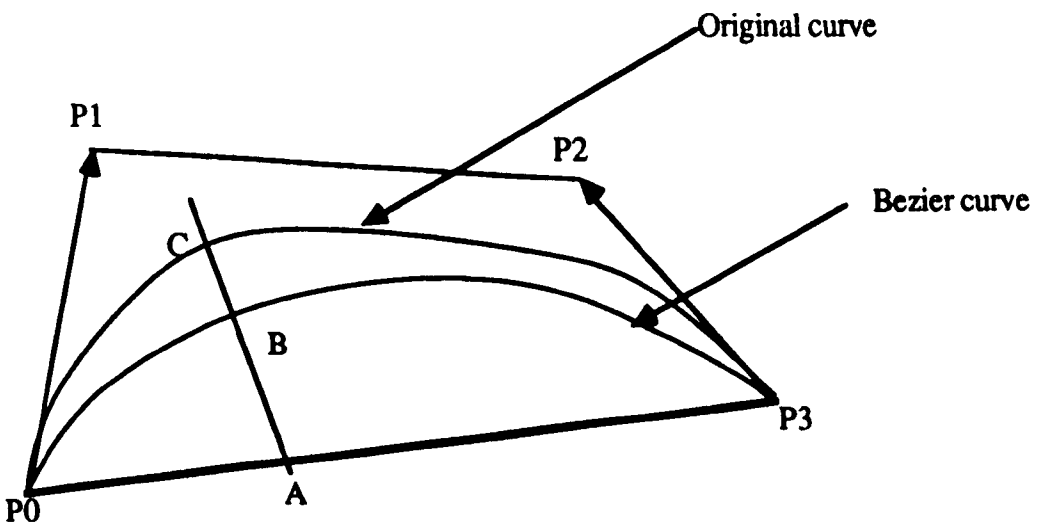


Fig.7.12, illustration of maximum error

Having got the magnitude of error, CB, we can adjust P0P1 by at most CB. Should we adjust by CB, the resulting maximum error will be always less than the distance CB, this can be shown as follows:

The bezier curve is specified by

$$p(t) = (1 - t)^3P0 + 3t(1- t)^2P1 + 3t^2(1-t)P2 + t^3P3$$

Let the gradient lines P0P1 and P3P2 have respectively u, and v as unit vectors,

and let λ, ß be the respective magnitudes of P0P1 and P3P2, then

$P1 = P0 + \lambda u$

$P2 = P3 + ßv$

Substituting these values into p(t) above leads to

$$p(t) = (1 - t)^3P0 + 3t(1- t)^2(P0 + \lambda u) + 3t^2(1-t)(P3 + ßv) + t^3P3 \quad (7.22)$$

let λ be adjusted by $\Delta\lambda$ to $\lambda + \Delta\lambda$, and ß adjusted by $\Delta ß$ to ß+ $\Delta ß$,

then p(t) becomes

$$p_a(t) = (1 - t)^3P0 + 3t(1- t)^2(P0 + (\lambda + \Delta\lambda)u) + 3t^2(1-t)(P3 + (ß + \Delta ß)v) + t^3P3 \quad (7.23)$$

Taking (7.22) from (7.23) gives

$$D(t) = 3t(1 - t)^2\Delta\lambda\, u + 3t^2(1 -t)\Delta ßv$$

D(t) denotes the difference curve.

looking at the basis functions $3t(1 -t)^2$, and $3t^2(1- t)$, and working out their appropriate first derivatives, we can find that their maxima occur respectively at $t = 1/3$ and $t = 2/3$; so the maximum error in vector form is

$4/9\ (\Delta\lambda\, u + \Delta ß\, v)$ **(7.24)**

if we have two vectors P and Q, it is known that

$|P + Q| \leq |P| + |Q|$ (triangle inequality), using this (7.24) gives

$$|(4\Delta\lambda)/9\ \mathbf{u} + (4\Delta ß)/9\ \mathbf{v}| \leq 4/9\ (|\Delta\lambda| + |\Delta ß|) \qquad (7.25)$$

This result proves our claim that, in general, if the magnitude of the tangent lines changes by ΔM, the maximum absolute error resulting from the fitting would be less than the absolute value of ΔM. This new result is very significant because it speeds up the fitting process, in the sense that if the correct estimate of the maximum error is available, it is applied directly with the knowledge that the resulting error would not be greater than the previous error.

Since the fit needs only be of a certain accuracy, the iteration is stopped if the change in gradient magnitudes is small enough (< 5 %) or a sufficiently small value of gradient magnitude is reached (< 0.5). Also, if a negative value of gradient magnitude occurs, the magnitude is set to zero and the iteration is stopped.

The limitation of this technique is that continuity of curvature is not catered for. Techniques exists for providing curvature continuity Bezier curves, but this may be done at the expense of more computing time. One other limitation is that the effect of adjusting a tangent magnitude propagates throughout the entire bezier curve.

Figures 7.13, to 7.16 show how good the approximations are.

In these figures, looking clockwise from the bottom left to the bottom right we have :

1. The original hand generated trace.
2. The Bezier cubic approximation.
3. The Bezier cubic approximation with relevant points (i.e Bezier polygon).
4. The Bezier cubic approximated and original traces are overlayed for visual comparison

For example, in figure 7.13 we have :

Fig.7.13.a is the original hand generated trace.

Fig.7.13.b is its Bezier cubic approximation.

Fig.7.13.c is the Bezier cubic approximation with its generating polygon.

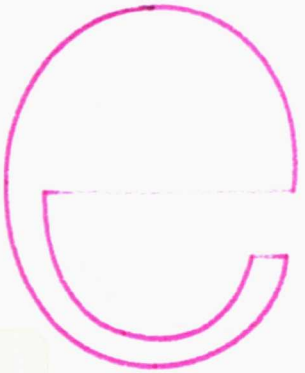
Fig.7.13.d is the result of putting the approximated pen trace on top of the original for visual evaluation.

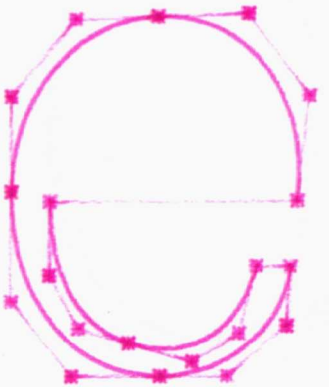
In these approximations (figures 7.13 to 7.16) we have arranged to have all representations achieve the same accuracy according to the maximum error. The bezier approximation requires substantially fewer selected points. Very close examination of the approximation does indicate that the cubic bezier approximation is the most accurate according to the maximum error norm; there are fewer gaps in the overlayed curves.

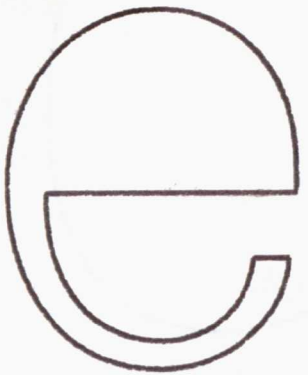
Having discussed the qualitative evaluation, let us now look at the the quantitative evaluation in general.

Quantitative evaluations

We compare the two representations by looking at their performance, i.e compactness, reliability, efficiency while accuracy remains fixed. By looking at a whole range of accuracies, we will observe the overall performance of the representations. Accuracy, unless otherwise stated, is measured in terms of







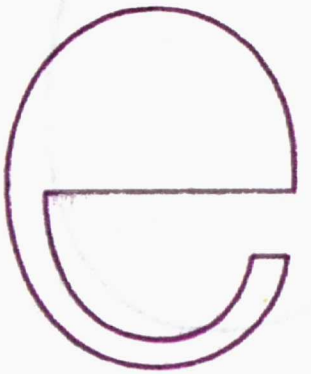
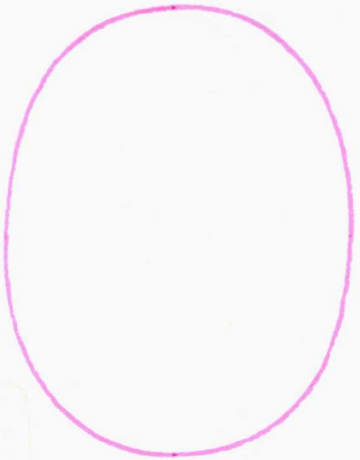
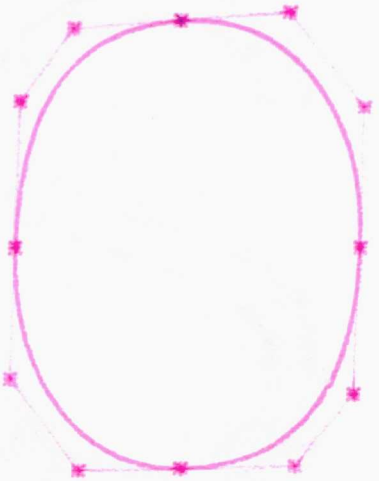
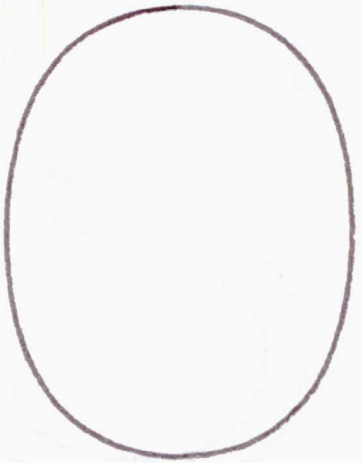


Fig.7.13
Clockwise from the bottom left to the bottom right, we have:
Fig.7.13.a : original trace.
Fig.7.13.b Bezier cubic approximation
Fig.7.13.c Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.13.d Bezier cubic approximated and original traces overlayed.







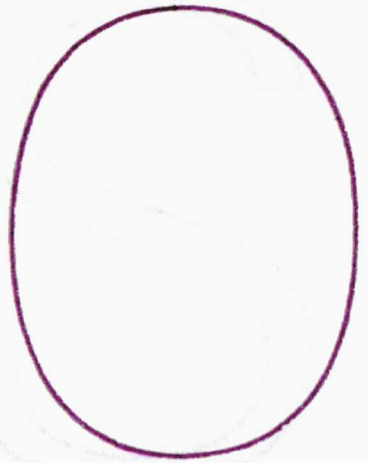
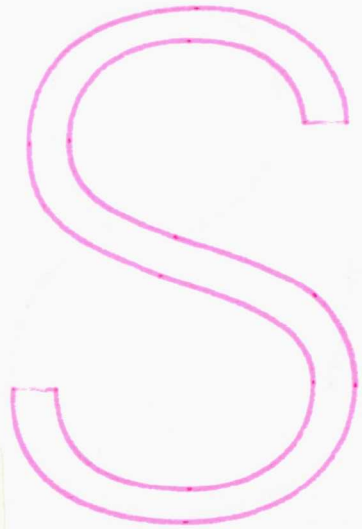
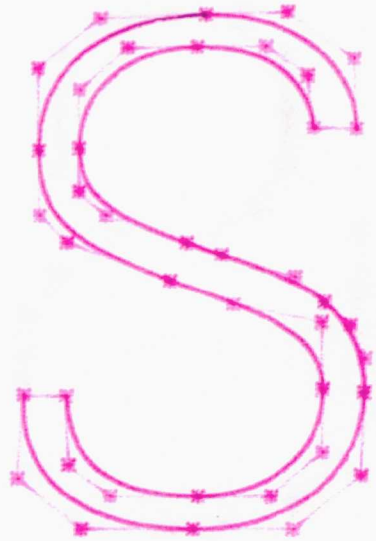


Fig.7.14
Clockwise from the bottom left to the bottom right, we have:
Fig.7.14.a : original trace.
Fig.7.14.b Bezier cubic approximation
Fig.7.14.c Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.14.d Bezier cubic approximated and original traces overlayed.





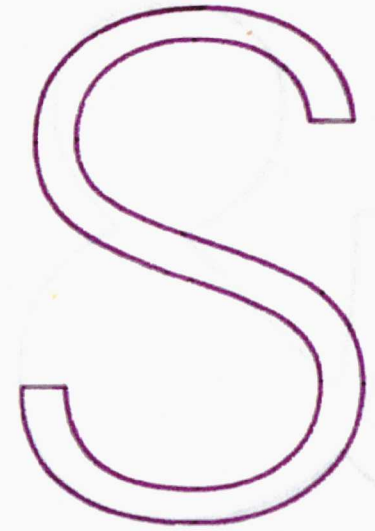
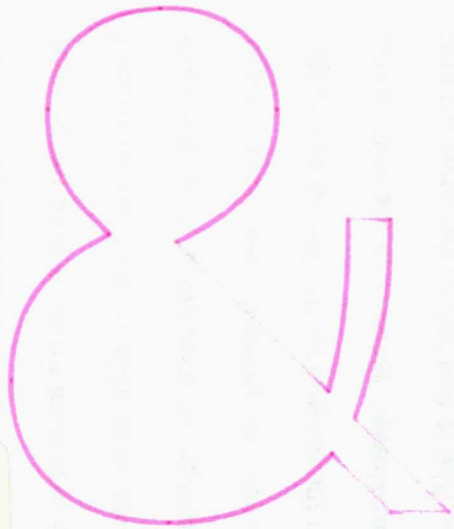
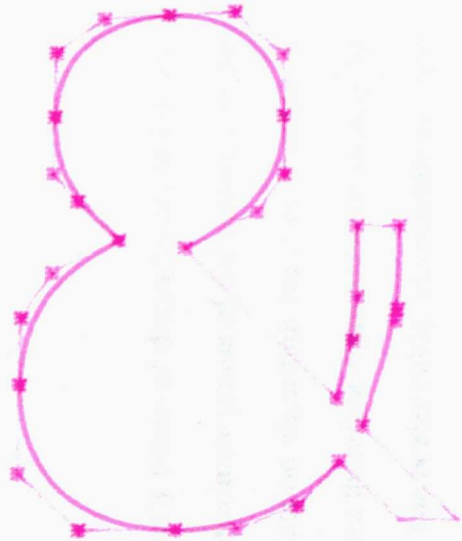
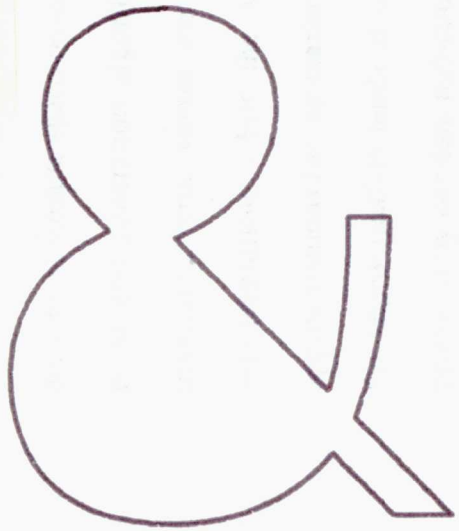


Fig.7.15
Clockwise from the bottom left to the bottom right, we have:
Fig.7.15.a : original trace.
Fig.7.15.b Bezier cubic approximation
Fig.7.15.c Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.15.d Bezier cubic approximated and original traces overlayed.







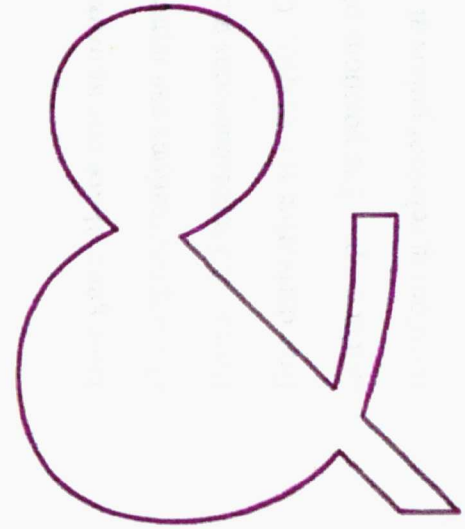


Fig.7.16
Clockwise from the bottom left to the bottom right, we have:
Fig.7.16.a : original trace.
Fig.7.16.b Bezier cubic approximation
Fig.7.16.c Bezier cubic approximation with relevant points (i.e Bezier polygons).
Fig.7.16.d Bezier cubic approximated and original traces overlayed.

maximum error estimated as explained previously.

Our algorithms involve a prescribed tolerance on maximum error. In selecting relevant points, our algorithms calculate maximum error heuristically.

The actual maximum error is guaranteed to be less than the tolerance. So to compare our representations and algorithms in a fair manner, we have exhaustively run each of them on many samples of hand generated material for a large range of tolerance values. For each run, we have calculated the actual maximum error, the number of interpolating points (i.e relevant points), the amount of storage space required, and the execution times of both the relevant point and generation algorithms. We are thus able to plot the estimated maximum error versus the required storage or execution time for each representation. For the visual accuracy in general, we have found approximations having maximum errors in the range 1 to 30. Below that range, the approximation tends to consume more storage space than the data points produced by our data reduction algorithm. Above that range, the approximated shape is often so distorted, as to be unrecognizable. As our digitizing tablet resolution was 396 by 522 units, a maximum error of 3 corresponds to an error of 0.5 % which is quite good. At this accuracy, one can seldom notice any visual differences.

How general are our approximations ?

Our approximations can represent accuraretly points of discontinuity in the pen traces. All representations are able to approximate points of discontinuity in the first derivative (i.e cusps). Our data reduction algorithm has a mechanism to detect cusps. The positions of the cusps are therefore represented accurately by positioning relevant points at each cusp. The bezier cubic polynomials are also able to approximate the size, orientation, and shape of a cusp because of the control they have over local slope on either side of an interpolating point.

The Bezier cubic representation is able to represent accurately regions of high

curvature. The linear spline representation must use multiple, closely spaced relevant points to achieve the same effect. The third order polynomial representation can also accurately approximate points of inflexion whereas the linear representation must introduce many knots and derivative discontinuities into the pen trace. None of our representation has built into it any mechanism to cater for curve crossings, but we have to say we have not noticed any problems related to this. Usually some other feature (e.g corners, cusps or discontinuity) is associated with important crossing or junctions. Relative extrema are also specifically handled. But in general these are regions of relatively higher curvature than the surrounding regions and so are adequately maintained.

Our generality criterion also contains a requirement for local smoothness. That is a representation should neither introduce any new feature nor remove any existing feature from the original curve. Upon looking at the shape examples described by cubic bezier and straight line approximations, we must conclude that the straight line approximation does not obey this criterion. It almost always introduces kinks in the approximation where the relevant points are located. They have been added by the representation. The cubic Bezier representation is able to adapt quite well to all local variations in smoothness.

How accurate and efficient are our representations ?

We will evaluate experimentally the efficiency of our representations and algorithms. The first type of efficiency to consider is that of transforming the raw data points into the representation. This in turn, can be split into two parts: the efficiency of the knot selection algorithm and the efficiency of the representation generation algorithm. These two efficiencies are discussed for each given piecewise polynomial approximation:

Straight line approximation case:

Figures 7.19, 7.20, and 7.21 are plots of accuracy versus selection time, accuracy

versus storage and tolerance versus accuracy, respectively, for the straight line approximation technique discussed in chapter 5.

The attained accuracy is used as the abscissa in these plots.

As far as efficiency is concerned, as is to be expected, Fig.7.19 shows that our sequential algorithm, requires smaller execution times for less accurate approximations (i.e for larger errors); in other words as is to be expected, Fig.7.19 shows that as accuracy decreases so does the time required to generate the representation. The algorithm has complexity O(npoints), because it uses a scan along selection mechanism which examines every data point.

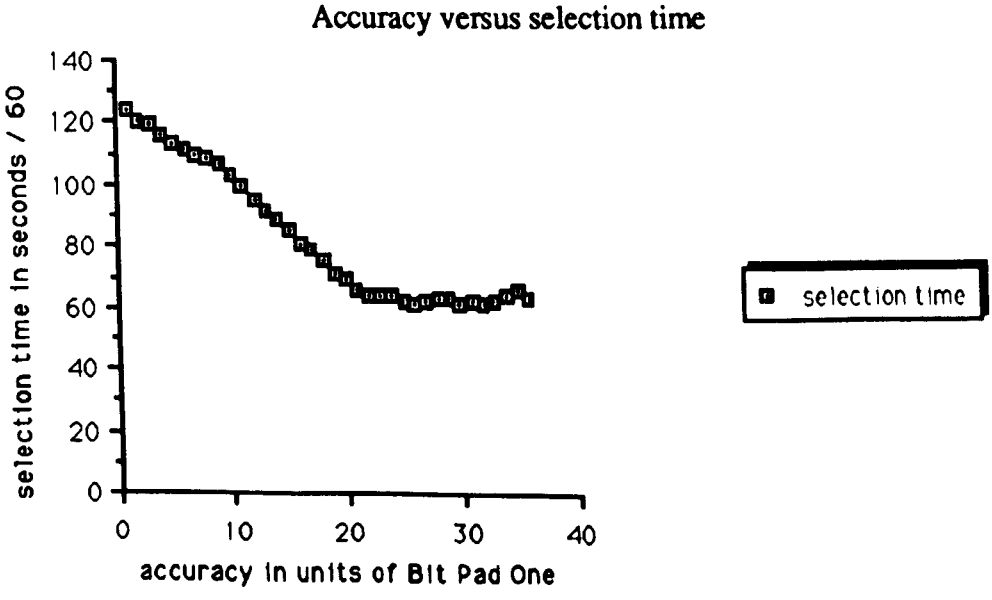


Fig.7.19 Accuracy versus selection time for linear approximation

Regarding storage saving (or compactness) we see from Fig.7.20 that we have an almost exponential trend. The plot shows that substantial savings in storage can be allowed for maximum errors greater than 3. At this accuracy, we have found that the approximated pen traces do not visually differ significantly from the original.

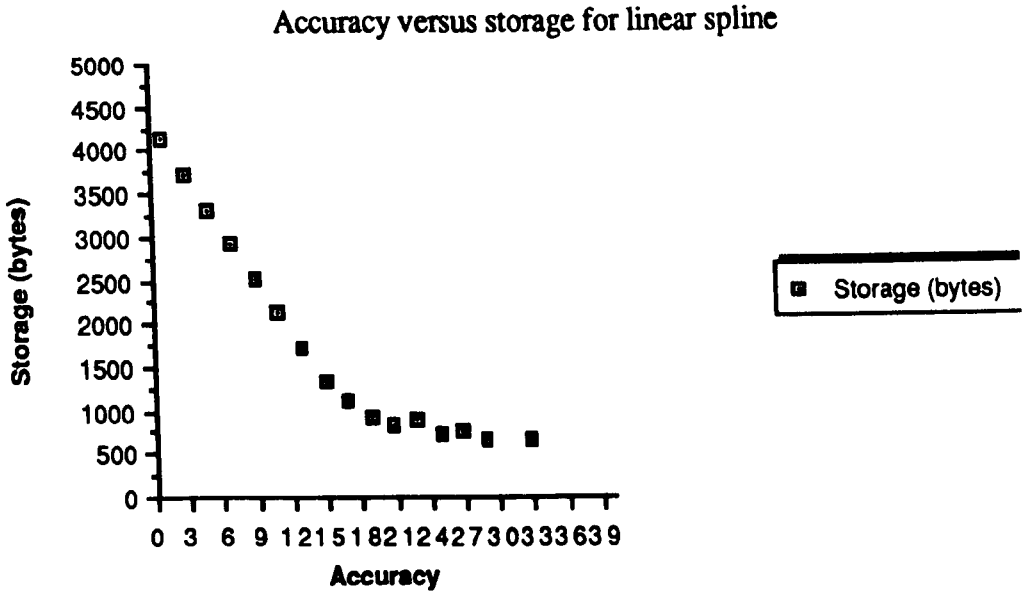


Fig.7.20 Accuracy versus storage for linear approximation

When we plot the specified tolerance against the estimated accuracy, we obtain Fig.7.21.

The experimental results seem to differ significantly from the theoretical results portrayed by the line equation ($t = \alpha$), where t is the tolerance and α the accuracy. The experimental graph indicates that $t = 2\alpha$. We can note that the plot may help us improving the reliability of our algorithm.

If τ is a user prescribed tolerance, then an algorithm should internally work with tolerance τ_a, say, we should have $\tau_a = \tau/2$ for our algorithm. In an actual implementation we would make more precise calculations. It appears that α is a linear function of τ,

$\alpha = a\tau + b$. From the results produced by our algorithm we can determine a and b using, for instance, least square fitting and set $\tau_a = (\tau - b)/a$.

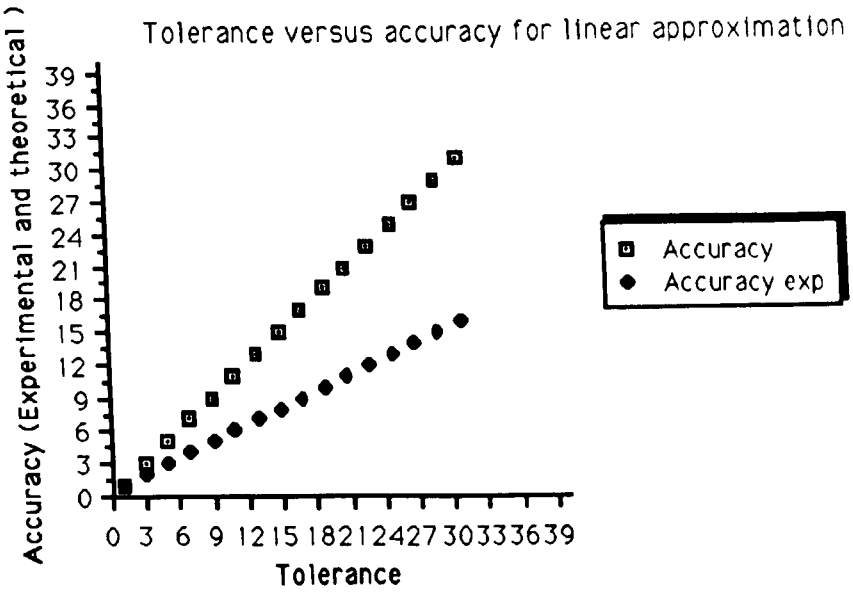


Fig.7.21 Accuracy versus tolerance for linear approximation.

Bezier cubic approximation case:

Efficiency, compactness and reliability of relevant point selection for bezier cubic polynomials can be evaluated using figures 7.22 , 7.23, and 7.24 respectively.

Again, we see (Fig.7.22) that the time for point selection is a decreasing function of accuracy. We can see that the bezier cubic fitting process is slower than the straight line fitting process. However this slowness is compensated by smooth representations for parts of the trace which are supposed to be smooth.

Fig.7.22, indicates the decreasing trend of the fitting process for bezier cubic representation, is not as accentuated as that for straight line approximation.

In other words, the rate of decrease is slower.

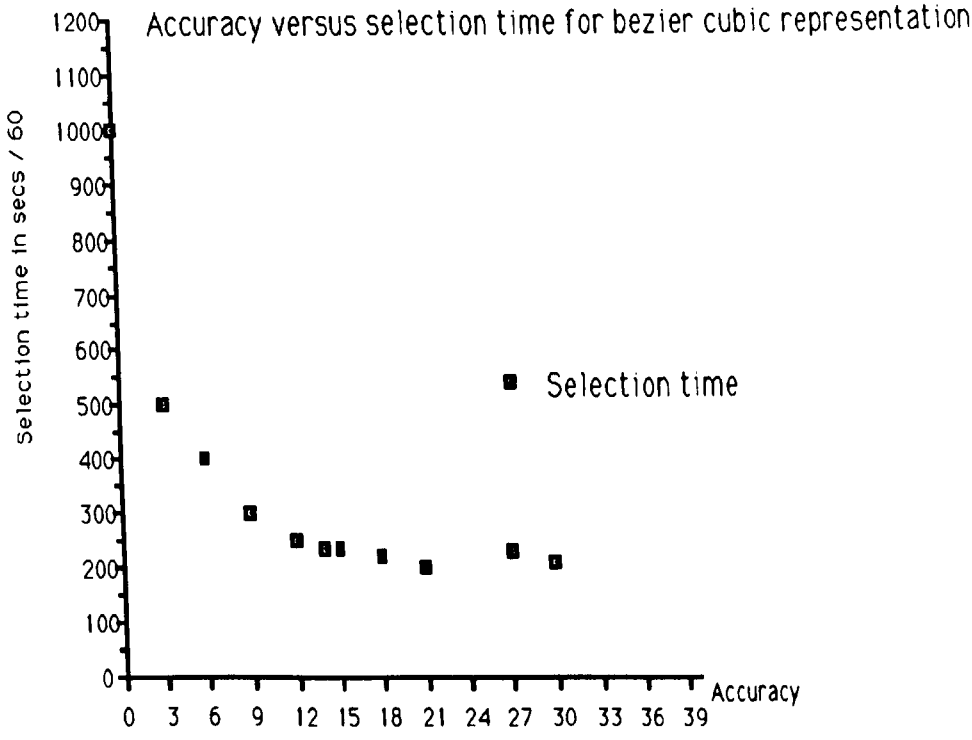


Fig.7.22 accuracy versus time for bezier cubic polynomial

Using Fig.7.23, we observe that bezier cubic polynomials require less storage than the linear splines. We can see, that for the same accuracy, the bezier cubic representation requires on average three times less storage than the straight line approximation.

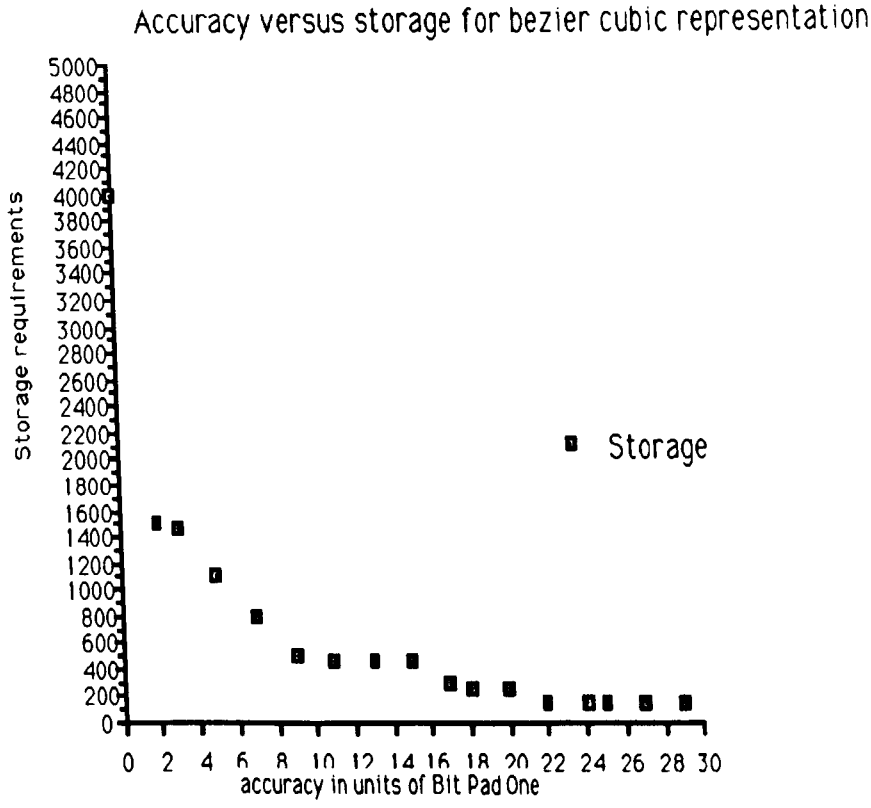


Fig.7.23 Accuracy versus storage for cubic bezier representation.

From Fig.7.24, we can say the fitting process performs much better for bezier cubic representation than for linear polynomials. We note that the idea of introducing an internal tolerance which is a linear function of τ applies also here.

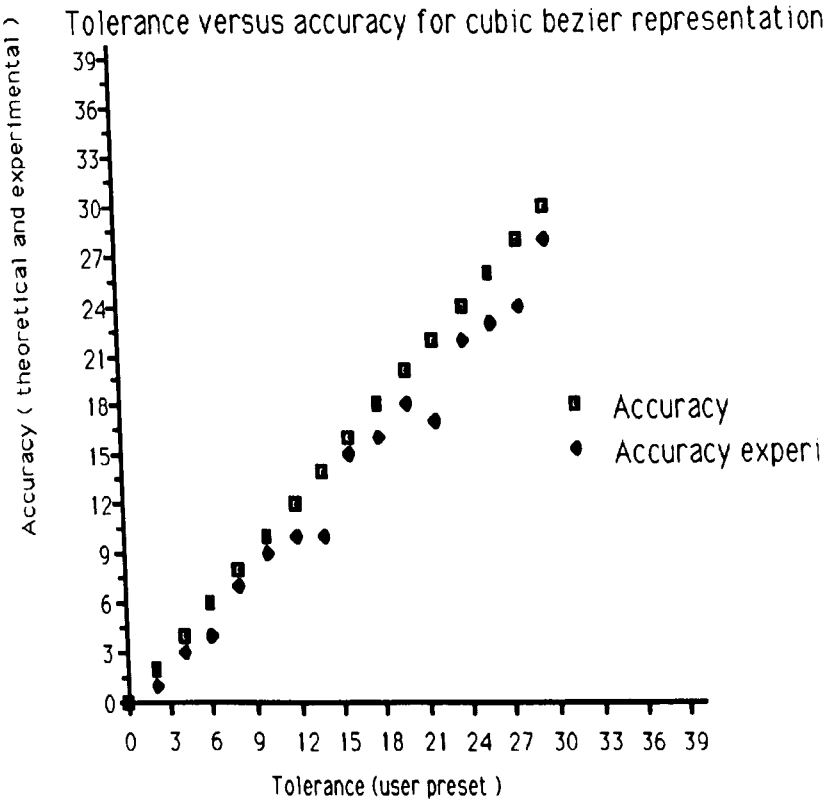


Fig.7.24 Accuracy versus tolerance for linear approximation.

Conclusions on straight linear approximation and cubic bezier approximation.

The linear and cubic bezier representations meet the compactness criteria quite well. However the cubic bezier representation produces much more compact representation than linear representation; but the price for achieving that compactness is that more time is required for the fitting: for a given tolerance, fitting cubic bezier curve, is much more slower than fitting straight lines.

The linear approximation does not meet the generality criterion in that it cannot match the local smoothness of many typical hand drawn shapes. Even with a smaller tolerance, and, therefore increased accuracy, this is still a major flaw; because of the obvious kinks for magnified pictures.

The cubic bezier polynomial representation meets the generality criterion quite well by representing the important features of shape and by being adaptive to the local smoothness of the shape. While the cubic bezier requires, in general, more execution time than the corresponding straight line approximation, we are still dealing with times under 3 seconds to represent a typical hand drawn shape. The advantage we gain in terms of generality is well worth the extra time. Finally, we will also trade-off the extra time required to evaluate a cubic bezier representation over a linear spline representation for the increased generality and compactness we achieve.

7.4 Is it possible to use uniform B-spline as an interpolator ?

RIESEN73 discusses the theory of B-splines and shows that they have some unique features that are suitable for curve representation. Some of them are continuity in slope , continuity in curvature, local uniqueness, fidelity to the defining vertex polygon. Let V_i, V_{i+1}, V_{i+2}, V_{i+3} be ordered positions vectors then a uniform cubic b-spline curve segment $C_i(t)$ is usually defined as (FOLEY82)

$$C_i(t) = E_0(t)V_i + E_1(t)V_{i+1} + E_2(t)V_{i+2} + E_3(t)V_{i+3} \quad (7.26)$$

where $C_i(t)$ is the parametrized location of the segment i;

with

$$E_0(t) = -1/6t^3 + 1/2t^2 - 1/2t + 1/6$$

$$E_1(t) = 1/2t^3 - t^2 + 2/3$$

$$E_2(t) = -1/2t^3 + 1/2t^2 + 1/2t + 1/6$$

$$E_3(t) = 1/6t^3$$

$0 \leq t \leq 1$ for each segment and i varies from 0 to n-2.

We want $C_i(t)$ to pass through a sequence of the selected relevant points

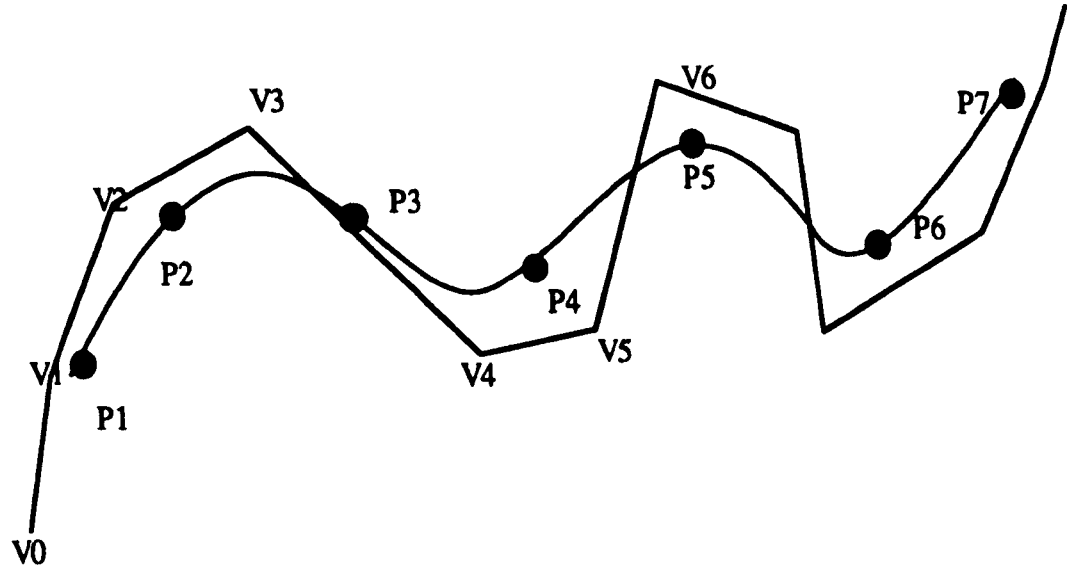
$P_1, P_2, P_3, ..P_j, p_{j+1}, ...P_n$

Looking at equatiom (7.26), for a segment i-1

$C_{i-1}(0) = P_i$; $C_{i-1}(1) = P_{i+1}$; this leads us to

$$V_{i-1}/6 + 2/3V_i + V_{i+1}/6 = P_i \qquad 1 \le i \le n \qquad (7.27)$$

Setting the conditions for the ends of the curve, we obtain a system of equation of n equations with n unknown $V_1, V_2...V_i$; as to the end conditions, we set $V0 = V_n$, $V_1 = V_{n+1}$ for the closed curves, and $V_0 = V_1$, $V_n = V_{n+1}$ for closed curves. The pictorial relationship between the vertices V_i and the relevant points is shown in the following figure



This method generally gives the same results as the bezier technique, with the only exception that, curvature continuity is ensured between parts of the pen trace which describe a smooth curve. As curvature appears not be a perceptual requirement, we did not find any major advantages of b-splines over Bezier curves. However there is a major disadvantage in the effective use of b-spline for sequential fitting. A sound fitting will require a prior knowledge of all the knots,

requirement, because all the original information about the pen trace, must be known before the fitting (i.e encoding) can be carried out.

7.5. Actual curve generation

Having specified a cubic (b-spline or bezier form), the problem the decoder has to face is that of calculating points on a cubic. Let a segment of the pen trajectory be described by

$$p(t) = at^3 + bt^2 + ct + d \qquad (7.27)$$

Varying the parameter t from 0 to 1 defines all the points along the curve segment. Considering equation (7.27). Calculating each point requires 6 additions and nine multiplications. This implies that (7.27) is computationally intensive. Equation (7.27) can be rewritten to use only six additions and six multiplications, i.e

$p(t) = ((at + b)t + c)t + d$. Although this technique economizes on computation, the multiplications required make it unattractive for efficient implementation. Here we are going to repeat the technique we use in chapter 5 for generating the straight line. Let us recall the generating procedure in the context of straight line; let the straight line be represented by $p(t) = at + b$.

Consider evaluating p(t) for n + 1 equally spaced values of t. Thus we want to

find $p_i = p(i\delta)$, where $\delta = 1 / n, 0 \leq i \leq n$.

The forward difference of a function p(t) is

$\Delta_1 p(t) = p(t + \delta) - p(t)$; we can rewrite this as $p(t + \delta) = p(t) + \Delta_1 p(t)$, this means

that we can determine $p(t + \delta)$ if we know p(t) and $\Delta_1 p(t)$. In iterative terms we have

$$p_{n+1} = p_n + \Delta_1 p_n \qquad (7.28)$$

for the straight line $\Delta_1 p_n = a\delta$; so we can evaluate p(t) in constant steps of size δ.

For a cubic polynomial $p(t) = at^3 + bt^2 + ct + d$, so the forward difference

$$\Delta_1 p(t) = a(t+\delta)^3 + b(t+\delta)^2 + c(t+\delta) + d - (at^3 + bt^2 + ct + d)$$

$$= 3a\delta t^2 + (3a\delta^2 + 2b\delta)t + a\delta^3 + b\delta^2 + c\delta \qquad (7.29)$$

We can see that $\Delta_1 p(t)$ is still a second order polynomial in t. This is unsatisfactory since evaluating (7.28) still involves evaluating a second order polynomial, plus an addition. We go one step further to apply forward differences to $\Delta p(t)$ to see if its evaluation can be simplified. Considering $\Delta p(t)$ as a function in its own right, we apply forward difference as

$$\Delta_2 p(t) = \Delta_1 p(t+\delta) - \Delta_1 p(t) \qquad (7.30)$$

$$= 3a\delta(t+\delta)^2 + (3a\delta^2 + 2b\delta)(t+\delta) + a\delta^3 + b\delta^2 + c\delta$$

$$- (3a\delta t^2 + (3a\delta^2 + 2b\delta)t + a\delta^3 + b\delta^2 + c\delta)$$

$$= 6a\delta^2 t + 6a\delta^3 + 2b\delta^2$$

We now obtain a first order polynomial in t, $6a\delta^2 t + 6a\delta^3 + 2b\delta^2$

Applying again forward difference we obtain

$$\Delta_3 p(t) = \Delta_2 p(t+\delta) - \Delta_2 p(t) = 6a\delta^3 \qquad (7.31)$$

Now sumarizing the forward differences worked out so far :

$$\Delta_1 p(t) = \Delta_0 p(t+\delta) - \Delta_0 p(t)$$

$$\Delta_2 p(t) = \Delta_1 p(t+\delta) - \Delta_1 p(t)$$

$$\Delta_3 p(t) = \Delta_2 p(t+\delta) - \Delta_2 p(t)$$

where $\Delta_0 p(t) = p(t)$; we write the above forward differences in the following forms :

$$\Delta_0 p(t + \delta) = \Delta_0 p(t) + \Delta_1 p(t)$$

$$\Delta_1 p(t + \delta) = \Delta_1 p(t) + \Delta_2 p(t)$$

$$\Delta_2 p(t + \delta) = \Delta_2 p(t) + \Delta_3 p(t)$$

With reference to (7.28), we can rewrite the forward differences using index n

$$p_{n+1} = p_n + \Delta_1 p_n \qquad (7.31)$$

$$\Delta_1 p_{n+1} = \Delta_1 p_n + \Delta_2 p_n$$

$$\Delta_2 p_{n+1} = \Delta_2 p_n + \Delta_3 p_n$$

To use the forward differences in an algorithm which iterates on n from n = 0 corresponds to t = 0.

Thus, the initial conditions can be expressed p_0, Δp_0, $\Delta_2 p_0$, $\Delta_3 p_0$ which are respectively

$$p_0 = d \qquad (7.32)$$

$$\Delta_1 p_0 = a\delta^3 + b\delta^2 + c\delta$$

$$\Delta_2 p_0 = 6a\delta^3 + 2b\delta^2$$

$$\Delta_3 p_0 = 6a\delta^3$$

So equations (7.31) and (7.32) are used to generate the curve segment.

The computationally intensive operations are done only once during the initialization process (7.32). Looking at equations (7.31), we can see that three additions will be generated n times, generating p_1 through p_n. It is clear that, although simple methods for evaluating cubic polynomials require several multiplications for each evaluation, the forward difference method requires only three additions for each point.

7.6 Discussions of the entropy rates.

The Bezier cubic approximations of many samples (i.e 13 tutorials) of hand generated data, led to a new set of samples. The average sampling rate resulted from the Bezier described pen traces was about 7 samples per second. The nth order relative frequency distributions of the difference samples, were measured, and used to calculate the nth order entropy rate estimates. The entropy calculations were carried out, as explained in chapter 3, and are tabulated as follows :

order of correlation (n)	Entropy H_n (bits)	Theoretical bit rates (bits/s)	Practical bit rates (bits/s)
0	14	98	110
1	12.14	85	98
2	11	77	87
3	9.28	65	76
4	8	56	68
5	6.85	48	57
6	5.56	39	48
7	4.6	32	42
8	4.12	29	35

Theoretical and practical bit rates.

The analysis of Δx and Δy showed that Δx and Δy were in the range -128 to 127. As, there were exactly 256 different classes of Δx and 256 different classes of Δy, the zero order entropy of Δ was 14 bits.

Given an average sampling rate of 7 samples /s the corresponding entropy rate was 98 bits/s. This is over 51 % below the target bit rate of 200 bits/s.

The first order approximation H_1 consisted of putting in the correct Δ probabilities ; the information was then calculated from

$$H = H_x + H_y$$

$$H_d = - \Sigma\ p(\Delta)\log(p(\Delta))$$

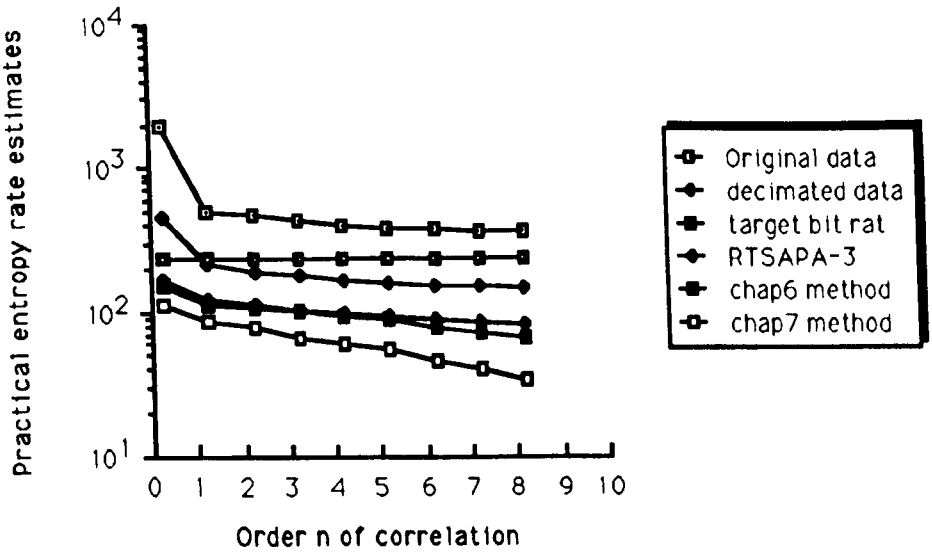
with $-128 \leq \Delta \leq 127$ in x and y direction; d is either x or y.

The second order approximation H_2 involved using conditional probabilities over pairs of adjacent Δ, calculating the information from the formula discussed in chapters 2 and 3.

The nth order approximation H_n involved using conditional probabilities over n consecutive Δ.

The last column of the above table, shows the practical limits of entropy rates. We can see that the first order practical estimate of entropy rate is 51 % below the target logic bit rate of 200 bits/s. The eighth order practical estimate of entropy rate is over 82 % below the target logic bit rate of 200 bits/s.

The graphical presentation of the pratical estimates of entropy rate, discussed in chapter 3 through 7, is shown in the following figure. The bottom curve curve portrays the measurements discussed in this chapter. If we compare all entropy rate curves with the target bit rate curve, there is no doubt that the entropy rate estimated from the Bezier cubic approximated data, is the winner, because the very low bit rate for handwriting and drawing signals can allow the combination with speech signals for simultaneous transmission over a single telephone circuit.



Entropy rate results pictured together for comparisons.

7.7 Conclusions

The new data reduction algorithm that was developed is applicable to any two dimensional shapes, described by digital functions $x(n\Delta t)$, $y(n\Delta t)$. The method can be used for various aspects of computer aided design, pattern recognition and computer graphics.

We have shown that linear approximation, is not really suitable for trajectories of the pen which are supposed to be smooth. We have found that an automatic fitting of cubic bezier curves to the pen trajectories produces acceptable curves in the context of smoothness criteria and compactness criteria. We have shown that intelligent fitting of bezier curves to pen trajectories provides highly compact data, from the point of view of storage and transmission; so compared to linear splines, cubic bezier curves give us the largest storage reduction. Experience has shown that reconstructed pen trajectories are almost indistinguishable by human eye from their original counterparts. That good quality reproduction can be obtained by storing a list of bezier points whose length compared to the list of digitized points is 2 % for simple shapes to 30 %

for complex handwriting.

Considering our qualitative criteria (generality), we consider cubic bezier curves to be more generally suitable than linear splines. Recall that generality can be thought of as smoothness as far as our visual perception is concerned. Linear splines are not considered general enough because they introduce discontinuities in the first derivatives - a very visible feature, this was illustrated in the previous chapter and the above section on linear approximation.

The relative frequency distributions of Δx and Δy derived from the Bezier cubic approximated data, have led to the lowest possible entropy rate of handwriting and drawing signals of 35 bit/s. This is over 82 % below the target bit rate of 200 bits / s .

8. REVIEW AND COMPARISON

In this chapter, we compare the various coding/decoding methods that are made possible by combining the techniques presented in the foregoing chapters. Error measures, both quantitative and qualitative, are further analyzed. Other possible techniques such as chain encoding are mentioned.

8.1 Further analysis of measuring errors incurred during pen trace segmentation

To be able to compare any two coding methods, error measures must be introduced to give a basis for comparison. These can be strictly quantitative, such as maximum deviation, or strictly qualitative, such as legibility. If the end purpose of a coding system is to transmit graphical material without loss of content, the visual criteria would seem to be more important. However, quantitative error measures are mathematical entities which are rigorously defined; for this reason the quantitative error measure was used throughout chapter 4 to chapter 7. Among the criteria for data fitting techniques, which were generality, accuracy, efficiency, reliability, convenience, and compactness, the quantitative measure applies to the accuracy criteria.

To obtain some sort of error figure, be it maximum error or rms error, a distance measure for two curves must be defined. This is not obvious, because there is no one-to-one correspondence between the run of points which make up the original pen trace and the run of points which make up the mathematically fitted curve. Thus some sort of dynamic distance warping algorithm is required. For our application, we try to map the generated points to the original points.

The algorithm works by trying to match every reconstructed point to some original point through minimization of the distance between the two points. The search for a minimum continues until some threshold difference is

exceeded. This thresholding allows for a minimum in the presence of quantization noise, but hopefully prevents the search from going around too far and missing a whole loop.

Choosing the threshold is rather heuristic, based on half the step size of the output data. If we run a strictly subsampled file through the error measure, the resulting error is null; this is because each point in the output file corresponds exactly to some input point. To overcome this problem, mid points of the reconstructed line segments are also used in the error measure. Lastly, the rms error measure for almost all investigated methods levels out at around 2 units expressed in terms of output device resolution.

Having established a quantitative technique that works well, let us pass to more subjective criteria. Both the rms and maximum error measures can be used. The maximum error is more meaningful visually, but is also more sensitive to one-shot errors. Other types of noise, such as high frequency jitter are more annoying visually than their error measure would indicate. This is is due in part to the sensitivity of the visual system to high frequencies. This also applies to the loss of features, such as the rounding of corners. Other features that are important include the closure of pen trajectories, accurate start and end points, and curve crossings.

Since it is impossible to state these considerations stringently, a few guidelines should be set. A good coding system should:

a) start and end accurately

b) have reasonable absolute error

c) reproduce corners well

d) not introduce extra wiggles or overshoots

These desired properties will be considered when comparing various methods.

Before starting the comparisons, a comment should be made on the classes of material being coded. The performance of a coding technique depends to some extent on the type of curve being processed. Smooth curvy drawings such as script writing and free hand drawing should produce different results than choppy pictures such as print and schematic diagrams. An attempt was made to test each algorithm on members of each class.

8.2 Discussions of techniques

Of those techniques that have been presented previously, some of the better ones are compared here. Here the analysis is restricted mostly to discussing these in qualitative terms. As mentioned previously, quantitative error measures are of limited usefulness in comparing techniques. For this reason, they are used as guides, and the qualitative properties given, previously are also analyzed.

8.2.1 Spatially sub-sampling technique

Subsampling is a good approach due to its simplicity and robustness. Test runs were made using various subsampling rates. The linear interpolator was the reconstruction method.

The performance of the subsampling technique degrades gracefully as greater compression is introduced. The error, however, is very dependent on the original drawing speed. Subsampling operates over a very wide range of compression factors. It works best for low to medium compression, but at higher compression it starts to chop off the corners and misses significant features. The rms error grows with the compression factor

8.2.2 Shape dependent segmentation

Three different segmentation techniques are included here:

The first fits a straight line in a sequential manner and uses the rms and maximum error as fitting guide.

The second fitting technique uses the notion that more points should be placed at the regions of high curvature and less points at the regions of low curvature. This second technique is the basis of fitting cubic bezier curves to successive segments of a pen trace. The corners are catered for at the expense of extra data points, and this may lower compression on some pictures made mostly of curves produced from sharp turning of the pen. The three techniques operate over a limited range of compression, one that is usually higher than that of subsampling. The number of points needed to represent a given symbol is independent of its size and drawing speed. This yields widely varying compression figures, but ones which are in some sense maximal with respect to tolerable error.

The shape dependent segmentation technique with corner detection is used as the basis for automatically fitting cubic bezier to pen trace segments.

Our bezier cubic fitting method generates curves which have the following properties :

1. Given a pen trace which does not have any corners or cusps, the fitted piecewise curve is tangentially continuous throughout.
2. A segment joining two consecutive points has no inflexion points if the segment emanates from the endpoints on the same side of the connecting chord; otherwise it has one inflexion point.
3. The tangent direction at a non corner point is determined chiefly by the position of the point relative to the four adjacent points.
4. The measure of the relevancy of a point must be close to or equal to the shortest euclidian distance from the point to the fitted curve.
5. The computations involved in establishing the relevancy of a point must

be relatively inexpensive.

Tests runs were made with an algorithm that uses one point on each side of the selected relevant point to perform the fit, and one which uses two points on each side. Significant improvement was found in using two points on each side of the selected relevant point so that the first technique, though it is simpler, was rejected.

The results of this technique are visually pleasing, although unforeseen glitches sometimes occur. Fitting cubic increases the compression ratio, but the pay-off is more computations.

8.3 Discussions of entropy rate measurements

When we look at hand generated material, we all intuitively recognize information, but in concrete terms, exactly what is it ? How do we measure how much information a hand generated material contains ? Using SHAN48's information theory, we have tried to answer these questions.

Chapter 3 provided the framework of the information measurements.

The relative frequency distributions of appropriate features of hand generated material were used to estimate the entropy rates of the signal associated with the handwriting and drawing.

Pen movements (i.e successive coordinate differences) were found to be the appropriate features of handwriting and drawing, because:

1. They were highly correlated.
2. The amplitude of the difference signals was substantially correlated, for instance in chapter 3, we recorded 18 bits resolution for pen position and only 8 bits for relative pen position (i.e pen movements Δx, Δy).
3. The distributions of Δx and Δy were highly non uniform and followed an almost exponential trend. This is opposite to the distributions of absolute pen positions (x, y), which tends to be uniform.

4. The entropy rate of signal estimated from the distributions of Δx, Δy was lower than the entropy rate of signal estimated from the distributions of (x, y). This was due to the fact that the distributions of points (x, y) isolated on their own, tended to be uniform, where as the distributions of Δx, Δy were non uniform, and seemed to follow an exponential trend.

An analysis of entropy rate strongly suggested that reducing the data led to lower entropy rate estimates (see chapter 3). This prompted the investigations into data reduction techniques, which resulted in lower estimates of entropy rates and visually acceptable approximated pen traces. We have noticed (chapters 4 to 7) that the entropy rate estimate decreases with the degree of approximation of pen traces. In our work, the entropy rate is not meaningful if the approximated material is highly distorted, i.e does not convey the originally intended message. So we are not only concerned with the quantity; the quality is important as well. We think that one shortcoming of information theory is its restriction to the efficient transmission of messages without any regard for their meaning. Here we note that, efficiency is used in the context of bandwidth utilization. But in the framework of this thesis efficiency refers to lower bandwidth and visually acceptable picture. Saying that information theory deals only with the quantity of information and not its quality may be a valid criticism. However, SHAN48 carefully pointed out that the theory was concerned only with the efficient transmission of messages and not with their semantics.

Chapters 4 to 7 suggested that the higher the compression of data, the higher the first order entropy of Δx, Δy. But the entropy rate estimate is lower because the effective sampling rate decreases more rapidly than the increases in entropy. We rightly concluded that this was similar to the

behaviour of the channel capacity $C = B^*\log_2(1 + S/N)$ which suggests that if the bandwidth B decreases more rapidly than the signal to noise ratio S/N increases, we shall expect C to decrease substantially.

The entropy rate measurements are summarized in the following figure:

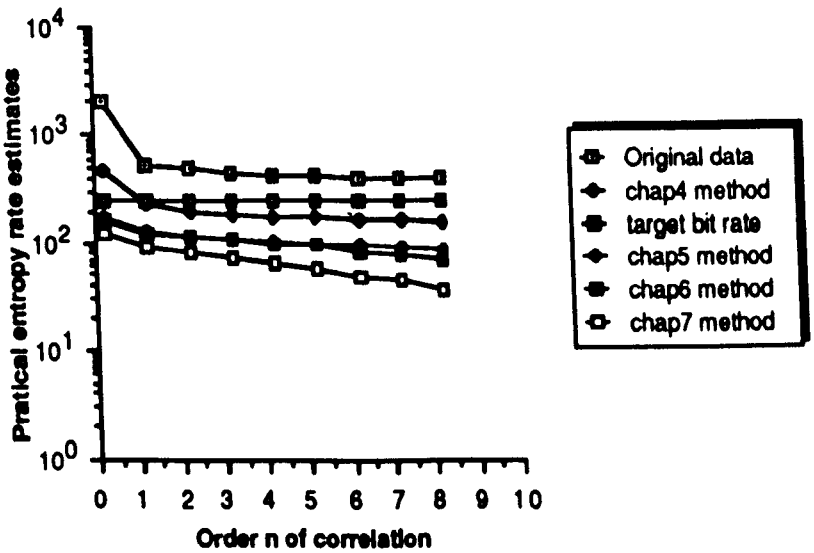


Fig.8.1 Graphical presentation of entropy rate measurements (bits /s)

In this figure the top curve is the graphical presentation of the entropy rate estimates of the signal. They are estimated from the relative frequency distributions of the original data. The 8th order entropy rate estimate was 314 bits/s, this is 57 % above the target bit rate. The bottom curve portrays the entropy rate estimates obtained from the distributions of data approximated by bezier cubic polynomials. The corresponding 8th order entropy rate estimate was 35 bits/s; this is over 82 % below the target bit rate of 200 bits/s.

Up to 8th entropy rate estimates were measured. It has been found that the higher the correlation the smaller the entropy rate estimates. From the second order correlation the difference between successive entropy estimates is small. The first order entropy rate estimated from the

distributions of the data approximated by methods in chapter 4 to 7 is below the target bit rate.

So it seems easier to construct a coding scheme based on first order statistics of approximated data and still meet the target bit rate requirements.

To approach the lowest possible entropy rate, a coding scheme may be constructed; but the price we pay is its complexity because it will have to exploit the 8th order statistics of the approximated data.

Given that RTSAPA-3 can operate in real time, and the first order entropy rate is 100 bits / s (see Table 5.7 of chapter 5); let us consider possible schemes for coding Δx and Δy.

8.4 Considerations for coding schemes.

In order to achieve a high data compression ratio, it is mandatory that each transmitted bit carries as much information as possible.

For our discussion, we use the distributions of Δx and Δy produced from the output of RTSAPA-3 (see Table 5.6 of chapter 5).

A study of Table 5.6 shows that Δx and Δy do not occur with equal probability. This is a source of statistical redundancy which can be reduced by a coding scheme, where short code words are assigned to frequently occurring coordinate differences (Δ) and longer ones to those occurring less frequently. The intuitive idea of assigning short code words to the most probable Δ and the longer code words to the least likely Δ, leads to a code with a small average length.

A study of Table 5.6 suggests that the distributions of Δx and Δy are quite similar. As the differences between those distributions are small enough, we can derive a mean histogram by pooling together Δx and Δy.

So the following table shows the mean distribution. Here Δ represents either coordinate difference in x or y direction. The Δ probabilities are

ordered by magnitude. The range of Δ was from -40 to 66. The set of Δ whose absolute value $|\Delta| > 10$ only made less than 2 % of the total number of Δ samples; and the probability of each of them was less 0.0005.

Δ	Relative frequency.
0	0.35660
-1	0.11887
+1	0.08915
-2	0.07132
+2	0.05943
-3	0.04430
+3	0.03531
-4	0.02935
+4	0.02511
-5	0.02194
+5	0.01949
-6	0.01752
+6	0.01592
-7	0.01458
+7	0.01346
-8	0.01249
+8	0.01165
-9	0.01092
+9	0.01028
-10	0.00970
+10	0.00919

Table 8.1 Relative frequency distributions of Δ from RTSAPA-3 output.

Variable length coding schemes

(e.g Morse code , Shannon Fano code, Huffman) have been around for over 40 years (HAM80). For our applications we will discuss Huffman coding procedure, which is known to be the most efficient variable length coding method.

In order to reduce the size of the lookup table for the practical implementation of both schemes, the Δ are partitioned into a frequent set and an infrequent set, termed "LFS" for less frequent set. We shall keep the discussions as brief as possible, since further details can be found in HAM80. The probability that "LFS" will occur must be kept small enough to maintain efficiency of the truncated Huffman encoding procedure. This probability will be smaller with the larger depth of the Huffman encoding,

Deltas source				Reduced sources
Deltas (Δ)	Probabilities	S1	S2	S3
0	0.35660	0.35660	0.35660	0.35660
-1	0.11887	0.11887	0.11887	0.11887
1	0.08915	0.08915	0.08915	0.08915
-2	0.07132	0.07132	0.07132	0.07674
2	0.05943	0.05943	0.05943	0.07132
-3	0.04430	0.04430	0.05446	0.05943
3	0.03531	0.04143	0.04430	0.05446
-4	0.02935	0.03531	0.04143	0.04430
4	0.02511	0.02935	0.03531	
-5	0.02194	0.02511		
5	0.01949			

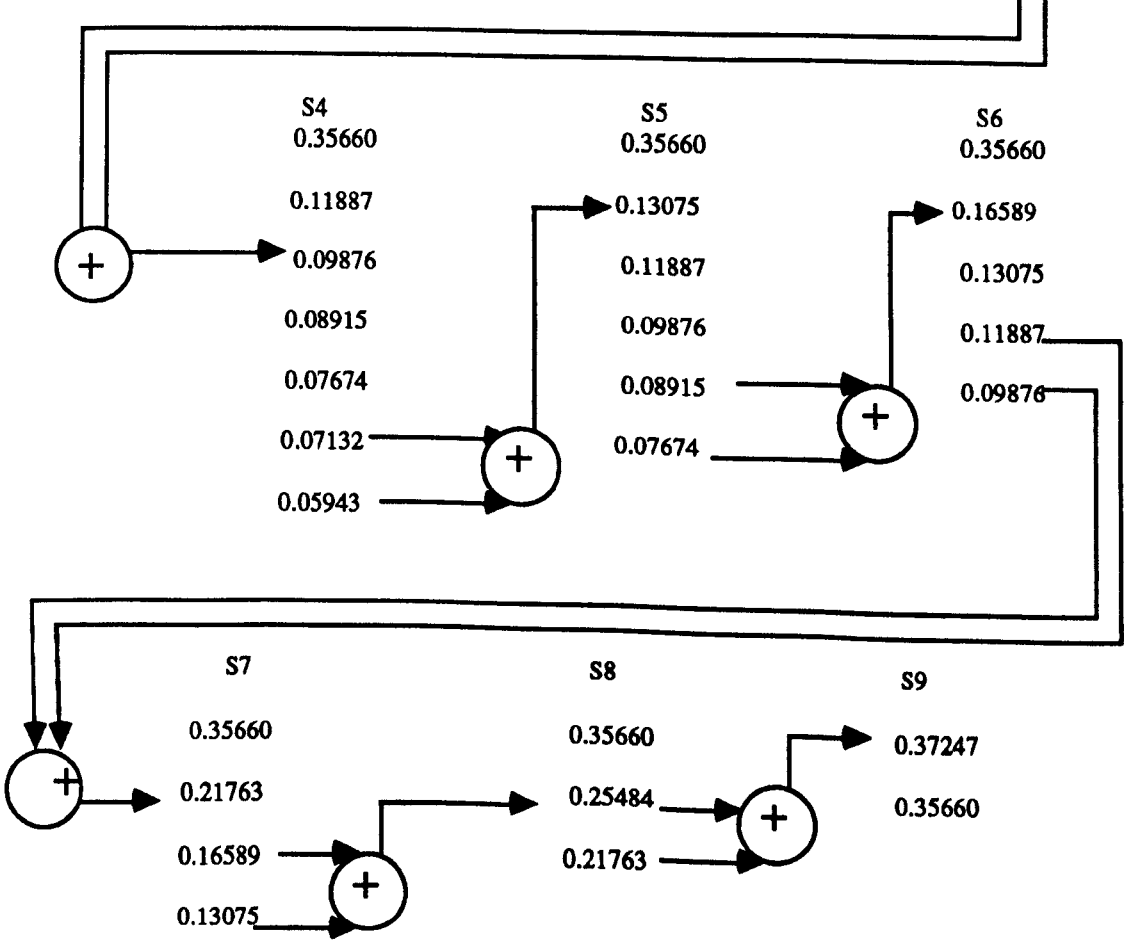


Table 8.2

which we define as the number of Δ's classes on either side of the central class (Δ = 0), that are employed for the construction of a Huffman code. We illustrate the Huffman procedure for depth 5. We first arrange the Δ's in order of descending probability as shown in Table 8.2.

Here are 11 classes of Δ's with probabilities

0.35660, 0.11887, 0.08915, 0.07132, 0.05943, 0.04430, 0.03531, 0.02935, 0.02511, 0.02194, and 0.01949 respectively. We now combine the two last Δ's into one Δ with probability $P_{10} + P_{11} = 0.04143$.

We now have 11 Δ's with probabilities

0.35660, 0.11887, 0.08915, 0.07132, 0.05943, 0.04430, 0.03531, 0.02935, 0.02511. These Δ's are rearranged in the second column in order of descending probability. We repeat this procedure by combining the last Δ's in the second column and rearranging them in order of descending probability. This is done until the number of Δ's is reduced to 2. These two reduced Δ's are now assigned 0 and 1 as their first digits. in the code sequence. We now go back and assign the numbers 0 and 1 to the second digit for the two Δ's which were combined in the previous step. We keep regressing this way until the first column is reached. The code finally obtained for the first column can be shown to be optimum (HUFFMAN52). The complete procedure is shown in Tables 8.2 and 8.3. A Huffman code is thus formed with all the Δ's in the first set. The truncated Huffman code resulting from a depth of 5 is read off the third column of Table 8.3. This code caters for all the Δ's in the first set. The infrequent set "LFS" is accounted for by introducing a special code word (0001) as a prefix to an ordinary 8 bit binary encoding of the Δ. The following table shows the code for an encoding depth of 5. Its average word length is 3.7 bits. Thus for an encoding depth of 5 an average code length of 7.4 bits is associated with Δx and Δy.

Deltas source			Reduced sources								
Deltas (Δ)	Probabilities	Code	S1			S2			S3		
0	0.35660	0	0.35660	0		0.35660	0		0.35660	0	
-1	0.11887	101	0.11887	101		0.11887	101		0.11887	101	
1	0.08915	100	0.08915	100		0.08915	100		0.08915	100	
-2	0.07132	1101	0.07132	1101		0.07132	1101		0.07674	1111	
2	0.05943	1100	0.05943	1100		0.05943	1100		0.07132	1101	
-3	0.04430	1000	0.04430	1000		0.05446	1001		0.05943	1100	
3	0.03531	11110	0.04143	11111		0.04430	1000		0.05446	1001	
-4	0.02935	11001	0.03531	11110		0.04143	11111		0.04430	1000	
4	0.02511	11000	0.02935	11001		0.03531	11110				
-5	0.02194	111111	0.02511	11000							
5	0.01949	111110									

S4		S5		S6	
0.35660	0	0.35660	0	0.35660	0
0.11887	101	0.13075	110	0.16589	111
0.09876	100	0.11887	101	0.13075	110
0.08915	1111	0.09876	100	0.11887	101
0.07674	1110	0.08915	1111	0.09876	100
0.07132	1101	0.07674	1110		
0.05943	1100				

S7		S8		S9	
0.35660	0	0.35660	0	0.37247	1
0.21763	10	0.25484	11	0.35660	0
0.16589	111	0.21763	10		
0.13075	110				

Table 8.3

Δ	Relative frequency	Code	Bit
0	0.35660	0	1
-1	0.11887	101	3
+1	0.08915	100	3
-2	0.07132	1101	4
+2	0.05943	1100	4
-3	0.04430	1000	4
+3	0.03531	11110	5
-4	0.02935	11001	5
+4	0.02511	11000	5
-5	0.02194	111111	6
+5	0.01949	111110	6
LFS	0.11161	0001 + 8 bits	12

Table 8.4 Truncated Huffman code of Depth 5 for Δ's generated from the output of RSAPA-3

Given the average word lengths of truncated Huffman codes constructed as explained above, the practical limits of the bit rates were subsequently estimated according to the method outlined in chapters 4 and 5. The following figure is a graphical presentation of the average bit rates against the encoding depth.

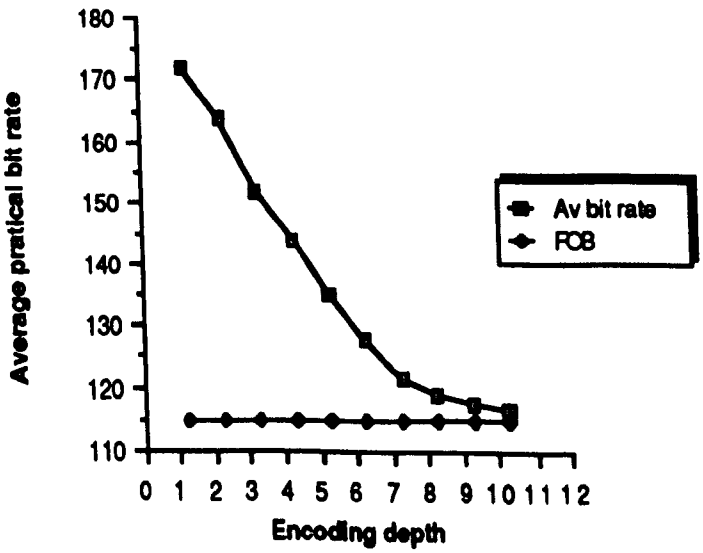


Fig.8.2 bit rate (bit / s) against encoding depth

Fig.8.2 shows 2 curves :

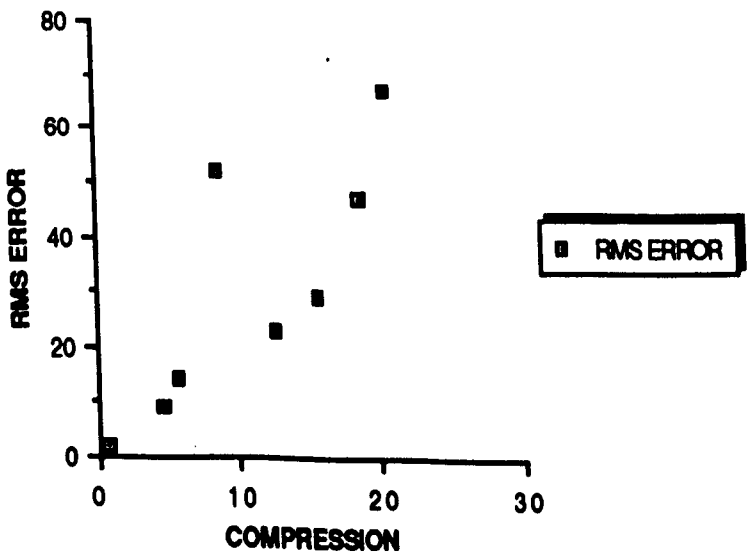
1. The horizontal curve labelled (FOB) indicates the average practical bit rate (113 bits/s) estimated from the first order statistics of Δx and Δy from the output of RSAPA-3 (see chapter 5).
2. The non linear curve portrays the average bit rates estimated from the average word lengths, including the contributions of the 8 bits suffixed attached to every code word "LFS" that could be achieved with truncated Huffman codes for encoding depths 2-10. The curve reveals that no substantial decrease of the average bit rate could be attained with encoding depths larger than 8. As expected, this curve confirms that the lowest achievable bit rate resulting from the Huffman code will be always greater than the practical entropy rate (113 bits / s) estimated from the first order statistics of Δ.

WITTEN87 believes that the state of the art in data compression is arithmetic coding, not the most celebrated Huffman procedure. WITTEN87 argues that arithmetic coding is superior in most respects to the better known Huffman method. Arithmetic coding represents information at least as compactly - sometimes considerably more. It encourages a clear separation between the model for representing data and the encoding of information with respect to that model. It accommodates adaptive models easily and is computationally efficient.

We have not investigated these claims, but they could be a potential issue for further research.

8.5. Summary of pen positions selection strategy

A comparison between subsampling and shape dependent segmentation is not easy since they operate under different assumptions. The shape dependent technique under consideration caters for corner detection as well, and yields better visual results for a small increase in data transmitted. Subsampling operates at a known compression ratio, and for a fixed frequency output. The number of points for any given symbol, and hence its legibility, will vary according to the writing speed, and hence to some extent according to the size. Shape dependent segmentation does the opposite. It generates a fixed number of points for any symbol, regardless of size. The resulting compression factor, or necessary data channel bandwidth, is not known ahead of time. For higher compression, this method is preferable. The technique is more sensitive to mis-segmentation error, this is indicated by the dispersed nature of the error versus compression graph depicted below



But these errors only affect a small neighbourhood. Automatic curve fitting has the same properties as the shape dependent segmentation on which it is based. Although it yields good fits, the extra data that must be sent cannot

be justified under normal circumstances from an error versus compression viewpoint. However, where a system already exists with the node and gradient representation as its primitive, this is not a problem. Also, the technique can be used to good advantage as a preliminary step in curve design, or in applications where maximal compression is not necessary.

As can be seen from the above discussion, the choice of method is dictated by its application, and not so much by the quantitative error versus compression performance. For our application, the shape dependent technique seems to be the most effective.

Our shape dependent techniques fit curve to data. When attempting to fit curves to data, the issues encountered were :

1. What should be the order of continuity between adjacent curve segments ?
2. What family of curves do we intend to fit ?
   This choice must be compatible with the continuity requirements and and must be amenable to reasonable error-criteria formulations.
3. Given our choice of curves, how do we decide to choose the interpolating points (i.e knots) ?
4. As discussed in chapters 6 and 7, we need to discover instances of certain features (e.g corners, straight lines) which play an important role in subsequent processing.

Ideally, our desire to preserve information requires that we preserve underlying discontinuities of all orders through the curve fitting stage. Also, new discontinuities should not be introduced. Practically, we have witnessed that these requirements are usually unfeasible, owing to noise in the data and the various unavoidable low pass operations the data may have undergone.

With respect to chapter 7, we have required our fitted curves to maintain position and tangent continuity, unless, discontinuities of these types were explicitly discovered. This choice was in part motivated by human sensitivity to such discontinuities. As regards to curvature discontinuities, we considered the empirical evidence in support of human sensitivity to them to be insufficient. Having decided on the order of continuity sought, we decided to use parametric cubic polynomials for a variety of reasons. First, they could satisfy our continuity requirements. Second, unlike with functions of an independent variable, we did not have to deal with awkward segmentation at points with infinite slope. Also unlike quadratic curves, cubic curves could cater for inflexion points. A good choice of knots usually prevented unnecessary wiggles in the fitted curve.

8.6 Do other well known techniques fit in ?

Other techniques exist that might be suitable for the real time coding of curves. Chain encoding (FREEMAN78) and predictive differential are two such possibilities, Since chain encoding is a popular and simple method, we discuss it :

Chain encoding consists of a linked series of incremental moves in one of eight directions (FREEMAN74, FREEMAN78). Thus, 3 bits are required for each move, and the accuracy of the method is determined by the grid mesh size. We conducted various tests with varying mesh sizes. The reconstruction method was simple linear interpolation.

The effect of the mesh size on the error is very similar to that of subsampling rate. It is difficult to compare this technique to the segment/interpolate schemes discussed in this thesis. Such a comparison would have to be based on a knowledge of the system resolution, sampling rate and output device size. Also, coding strategies would have to be examined. For example, for chain encoding we can pack two points per

byte. For node encoding, 4 bytes are usually required to send each coordinate pair (9 bits resolution). These types of consideration are best dealt with within the context of a particular application.

Thus, while other coding schemes exist for transmission graphical material in real time, they cannot be directly compared to the methods discussed in this thesis.

9. CONCLUSIONS

This chapter is a summary of the work presented in this thesis.

It reviews the " what, why and how" of our work.

It discusses the relative advantages and disadvantages of the coding techniques presented in the thesis. Recommendations are made about possible applications of the hand drawn coding curve coding and interpolating algorithms presented. We finish the chapter by pointing out areas for further research.

Summary:

The demands of the Cyclops system presented a practical motivation for doing work of much more general value. Under ITU auspices, our work falls in the field of telecommunication services called Telewriting.

Apart from the obvious educational role (e.g teaching at a distance) of the telewriting system, other applications are discussed in CCITT81.

From the system designer point of view, the existing coding strategies are discussed in CCITT85 (March and June 85), and our work is a significant contribution to that aspect of the telewriting system.

The entropy estimates provide a target for coding. We feel that the entropy a figure of merit for a coding technique. This is what is missing in most of work published to date (CCITT85). We felt that finding the minimum entropy of the handwriting and drawing signals. is a result of general scientific value which provided an additional justification for our work.

In order to measure the entropy of the handwriting and drawing signals, 13 one hour electronics tutorials were recorded. The recorded material was either one of the following types:

1. Handwriting only.
2. Drawing only.
3. Handwriting and drawing. This type was the most common.

The written medium of communication was English language. However, we felt that similar results would characterize hand generated material produced using any Latin based language. As the data representing hand generated material, "follows" line segments in time, we could claim that similar results could be obtained for any written medium of communication (e.g Japanese, Chinese, Arabic, etc..). But we have to confess and say that we did not conduct any experiments to verify this claim.

Initial standard estimates of the nth order entropy, based directly on conditional probabilities, led to results which were much higher than would be expected intuitively. The explanation appeared to be "unwanted detail" (noise, wobble, etc..). In order to test this hypothesis, we needed to find some way of fitting curves so as to reduce them to parameters that contain all the information a Human viewer needed in order to perceive adequate detail, while leaving out all the unwanted detail.

This representation contained all the relevant detail and we could, therefore, use it to extract the nth order (effectively limiting) entropy for the appropriately coded signal. This led us to a final result of 35 bits per second. This is 82 % below the practical target bit rate of 200 bits per second (CCITT81, CCITT85). The motivation for low bit rate is that it is possible to transmit graphic information and speech so that one can talk and write via the telephone line simultaneously. This means that a normal conversation can be accompanied by " handwritten " information in both directions of the telephone connection. The production of low bit rate (at most 200 bits per second) picture signal implies that a very small part of the telephone frequency band, the voice band (300 Hz to 3.4 kHz), is reserved for transmitting the pictures without any unfavourable influence on the telephone conversation. We may ask ourselves the following question: Why not use a very low bit rate such as 35 bits per second, which is

perfectly feasible (we have shown it !) ?

The answer stems from two conflicting requirements :

1. A very low bit rate implies greater telephone speech quality because only a very small band is extracted from the speech band.
2. The complexity of the filter (bandpass) design may be such that it may not be possible to achieve the graphic signal frequency range allocation.

So it makes sense to go for a bit rate which is just low enough to enable the following :

1. Acceptable speech quality.
2. Easier filter design.

We draw the attention of the reader to the fact that the use of error control coding (e.g Hamming codes; HAM80) for a very low graphic bit rate (i.e 35 bits / s) should produce a final higher graphic bit rate which permits an easier filter design. Such a graphic bit rate is 200 bits per second, because one can readily use the existing modems (e.g V21 or Datel 200, SMOL76) using F.S.K modulation. For the transmitting station the transmitting channel will operate on the tone frequencies are 980 Hz for bit " 1 " and 1080 Hz for bit " 0 "; and for the receiving station the transmitting channel will operate on the tone frequencies are 1650 Hz for bit " 1" and 1850 Hz for bit " 0 ". All that remains to do is to design two stop band filters.

Transmission station stop band filter characteristics :

LCF = 880 Hz, HCF = 1280 Hz.

Receiving station stop band filter characteristics :

LCF = 1550 Hz, HCF = 1950 Hz.

LCF and HCF represent respectively the low and high cutoff frequencies.

When these analog stop band filters are designed and constructed, they would be appropriately inserted between the lines which link the telephone set and the modem.

Chapter 9. 4

A major spin off of our investigation was the development of suitable techniques for coding hand generated material. A measure of effectiveness of these techniques was the entropy values they yielded.

The data reduction methods considered in this thesis were tested on point series sampled by Summagraphics Bit Pad (SUMG79). However we believe that our methods are applicable to planar point series sampled by any digitizing tablets, stereo-plotters or similar electromechanical devices in time-interval mode. The point series are captured because they carry description about the curves (i.e pen traces) from which they are sampled. The time sampling of the pen traces produced large amounts of data. Chapter 2 discussed the handling of large amounts of data, from the data compression point of view. This chapter points out that compressing data usually brings benefits such as :

a). Storage space reduction.

b). Reduction of transmission and processing costs.

Chapter 3 discussed the bit rates of digital hand generated material. It was found that the unexpected high bit rates were due to the noise contained in the data. Since sampling in time-interval mode was unintelligent, the degree of relevancy of the individual points varied over the point series. Therefore, it was shown that eliminating some of the points which carry the least relevancy should lead to low bit rates.

So subsequent chapters investigated ways to lower the bit rates in order to use effectively the narrow band channels (e.g. telephone channels).

Chapter 9. 5

In chapters 4 to 7 we have found that reducing hand generated data without losing too many relevant points, led to bit rates lower than the 200 bits/s. Such a low graphic signal bit rate makes the combination of handwriting and speech signals, possible for simultaneous transmission over a single telephone line (The why and how of this was discussed in chapter 1 and 2).

The methods examined for data reduction which ultimately led to low bit rates, have been of the segmentation / interpolation type. Thus selected points from the original pen trace are transmitted, possibly along with the derivative at these points. Interpolating algorithms are then used to reconstruct the original pen traces. The coding is done automatically without any user interaction. It was found that the most effective coding (i.e data fitting) technique was the cubic bezier polynomial and the regenerating technique (i.e decoding) was again derived from the bezier format. What is relatively new in our approach is that no human interaction is required, coding and decoding is done automatically, that is the difference between our work and most of the work published which usually involves the interactive methods to fit the curve. In our method, all the relevant points do lie on the generated curve, hence the use of interpolation. In our method of fitting cubic, the effect of any change in a relevant point or derivative is strictly local, confined to the two segments adjoining it.

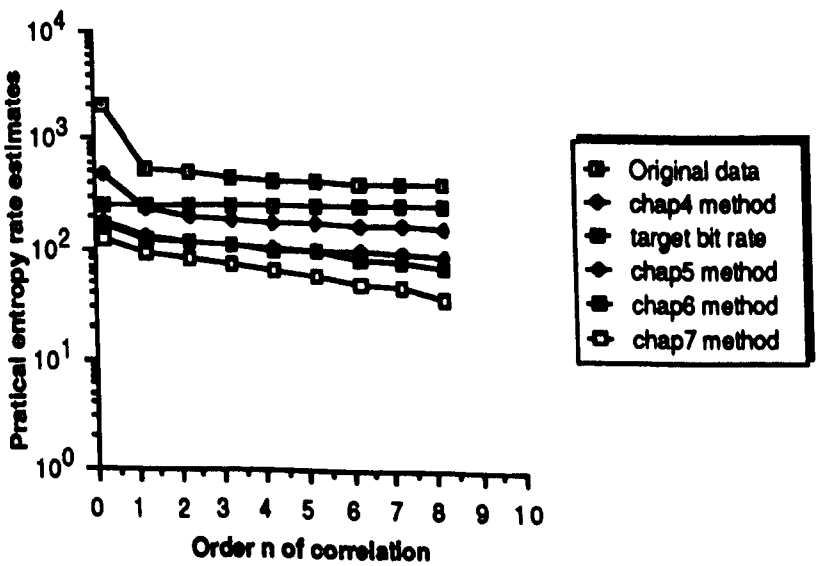
The use of cubic polynomials has other advantages:

1. The local control of the derivative allows for control of both the curve direction and curvature, thus giving good flexibility to the basis.
2. The calculation of the cubic segment is computationally simple when using the forward difference technique (DDA).
3. The representation is also fairly compact, especially when only the relevant points are transmitted; this, of course, increases the load of the decoder (e.g estimation of local gradient).

The results of chapter 7 suggest that a straight line based fitting method creates more relevant points than necessary, for a cubic interpolator. We can conclude that effective data reduction can be achieved only when there exits a complementary relationship between the data reduction algorithm and the subsequent interpolation algorithm.

How good were the entropy rate measurements ?

Chapter 3 caters for the entropy rate measurements obtained from the probability distributions of original data. The lowest entropy rate estimate was greater than the target bit rate of 200 bits/s. To achieve a bit rate less or equal to 200 bits/s, a practical analysis indicated that reducing data without introducing too much distortion (i.e visually acceptable) to the picture, was the answer. Hence chapters 4 through 7 investigated various reduction techniques. Entropy rate estimates were measured from the probability distributions of the reduced data. The results are graphically portrayed in the following figure:



The above figure indicates that the cubic fitting method discussed in chapter 7 is the most efficient because for visually acceptable and smooth pictures,

we get an entropy rate estimate of 35 bits/s in return. But approaching this bit rate may require a coding system which will make use of nth order relative frequency distributions of the reduced data.

As the order of correlation n = 8, for 35 bits/s, the required coding system may be too complex. However for n = 1, a bit rate of 98 bits/s is well below the target bit rate of 200 bits/s. So a simple coding scheme using only first order relative frequency distributions of the reduced data, would do. Such a low bit rate for handwriting signals, can allow the combination of handwriting and speech signals for simultaneous transmission over one single telephone line (The relevant technique was discussed in chapter 2).

Methodology for accepting a representation.

In order to determine the best trace representation, we established a set of criteria by which we could compare representations. The criteria were generality, accuracy, efficiency, reliability, convenience, and compactness using these criteria, we evaluated a series of pen trajectories, based on parametric piecewise polynomials. The cubic bezier polynomials was quite clearly the superior form. Visually accurate representations of pen trajectories, requiring only 2 % to 30 % of storage space allocated to the original data, can be obtained using cubic bezier polynomials. As well, the information is in a more tractable and higher level form for subsequent processing. The cubic bezier representation was found to be very general, and the fitting algorithm developed was quite reliable and convenient.

The first major part of the thesis was the introduction of several segmentation strategies. Subsampling is an existing method which under some circumstances still performs best. This the case when a fixed capacity channel is available for transmission of the coded message, and maximal compression is not needed. The error and fidelity of the coding depend on

the drawing speed of the user. This has both advantages and disadvantages, as seen previously.

Shape dependent segmentation methods based on local direction, discussed in chapter 5 through 7 are different in principle from the subsampling method. They largely remove the time and scale dependence from the coding process. They thus yield a constant relative distortion for a given symbol. In some sense, the compression is maximal for a given level of desired fidelity. The methods would be preferred where coding compression is important, and where the system is fast enough to accommodate the added complexity of the segmentation algorithm. They are also preferred where the material is to be stored and manipulated; this aspect is obviously attractive for educational institutions, such as the Open University which caters for teaching at a distance.

The automated cubic fitting has been dealt with in chapter 7, where the derivatives at the selected relevant points are determined from the local information present in the original trace. The gradients are adjusted to yield the best fit to the original trace. The algorithm is iterative. A limitation exists in the real time algorithm due to the inability to predict the next selected point derivative. An off-line iterative method has overcome this. Even with this limitation, the results obtained are quite good.

The second major part introduced the straight line fitting and the cubic fitting for pen traces. The third major part introduced various slope estimations for cubic interpolation.

In terms of coding efficiency, we found that methods requiring the sending of the significant points and slopes prevent them from being competitive with methods which require the sending of only significant points. In the case where the appropriate gradient representation already exists in the

system, or where the extra flexibility afforded by slope control is desired, the curve fitting technique then becomes attractive.

It should be noted that all of the techniques presented in our work, based on coding of sample points and possibly slopes, make the manipulation of the curves straightforward. Certain other techniques, such as chain encoding, do not allow for easy manipulation (i.e scaling), this is discussed in FREEMAN 74, FREEMAN78.

Techniques (CCITT81, CCITT85) which are incremental (chain encoding) or predictive (differential encoding) are also more sensitive to errors in transmission than straigthforward point encoding.

9.2 Purpose and implications of the work.

As mentioned in our introductory chapter, (chapter 1), during the inception phase of this thesis, our work proceeded with an engineering goal in mind. A primary application of our work is the enhancement of the Cyclops system; one of the aspects of Cyclops system is telephone conferencing, which allows for the interactive exchange of graphical material; Our proposed techniques could be used in the Cyclops system to efficiently transmit data which describe the trajectories of the pen moving on the writing surface of the digitizing tablet.

As we can see, the techniques discussed in our work have several possible applications; the first and primary of these is the real time coding of hand generated material. Depending on the system configuration, channel bandwidth and computational complexity permissible, several of the techniques presented in this thesis are usable.

A related application is the introduction of the cubic bezier as a new primitive in the O.U. Cyclops system. The flexibility afforded by such a primitive would increase the range of drawings that could be transmitted. As well, the end point representation ties in neatly to existing Cyclops

primitives such as lines and arcs. The selection of a relevant point, i.e interpolating point, and the associated gradient would allow for even greater versatility in the primitives. If such primitives were added to the system, then the standard Cyclops terminals could then be used as receivers for the interactive graphical communications application pointed out above.

The new data reduction technique (CRAMPING85) is a byproduct of this research. The data reduction algorithm is suitable for real time processing, and is based on the notion that to interpolate a curve efficiently, few points should be placed where the radius of curvature is large, but many where it is small.

Another byproduct of this research is the automatic fitting of cubic bezier polynomials to discrete two dimensional data. The cubic segment regeneration algorithm requires only additions and shifts, and the gradient is computed as a weighted sum of selected relevant points. The resulting algorithm is thus very fast.

We finish this section by saying that, the curve fitting techniques presented in our work can be used as a preliminary stage in curve design or computer graphic drawings composition. The ease of manipulation of this representation, and the flexibility afforded by local gradient control make it well suited for applications where precision is not of maximum importance. One such application could be in the information provider terminals which are used in composing database entries for Cyclops systems.

Very low bit rate transmissions over narrow band channels channels, e.g voice graded telephone lines are definitely possible, if techniques presented in this thesis are applied. Having achieved low bit rates for data, it is then possible to combine audio signals (i.e speech) and graphic signal for transmission over one single telephone line. From the economic point of view, this represents very substantial savings, in a telephone conferencing configuration, because instead of having two telephones circuits between the receiving end and the sending end; i.e one for graphic and one for speech as the existing Cyclops system, we only have one telephone circuit linking the two activity ends.

9.3 Is there any scope for further research ?

This thesis has established the basic theory and algorithms for achieving low bit rates transmission over narrow band channels. It has also introduced some segmentation techniques and automated two dimensional curve fitting techniques. Their primary use is for the real time coding of hand generated material.

Our work, however has stayed away from being implementation dependent. We feel that the evaluation of the techniques presented, against other coding techniques such as chain encoding must be conducted within the context of a given application. Some of the factors that would have to be considered include input and output devices, computational complexity of the algorithms and speed of available hardware, the nature and capacity of the data channel, effect of transmission errors, and coding efficiency versus simplicity and flexibility. A particular implementation would have to refine and simplify the algorithms, and also examine them for stability and error control. Specific coding conventions would have to be devised and implemented. As well, tests would have to be made in the actual intended working environment.

We feel that the main thrust of any subsequent development in this area would have to be implementation. There may be possibilities for the refinement of the segmentation and fitting strategies. Undoubtedly, there may be other approaches to these two problems.

9.4 Summary of conclusions

The information content (expressed in terms of the entropy rate) of the original data has been measured. Unexpected high bit rates led to the investigations of data reduction techniques, which may be used to lower the entropy rate. The straight line and cubic interpolators, which are standard methods in one dimensional curve (i.e waveform) fitting, have been applied to the problem of coding two dimensional shapes. In particular the Bezier cubic is used, and various new and old techniques for estimating the gradients are proposed and reviewed. The merits of these various techniques are compared. The pen trajectory segmentation necessary to generate the relevant points is examined next. The standard method of subsampling is examined next, and some relatively new techniques for shape dependent segmentations are discussed. Automated curve fitting is introduced as an alternative to determining the derivative solely on selected relevant points. Information from the original data is used to determine the gradient. The techniques were evaluated on many real tutorials. Emphasis was placed on the real time performance of the techniques examined; several of them yield good results and should be considered in those applications where such coding is to be implemented.

The lowest measured entropy rate estimate was 35 bits/s, which was possible from the 8th order distributions of the reduced data output by the bezier cubic fitting technique discussed in chapter 7. A first order distribution of the reduced data produced an entropy rate estimate of 96 bits/s; this is 52 % below the target bit rate of 200 bits/s.

Appendix

APCHAPTER5.1

The straight line equations are

$$x = t;$$

$$y = s*t;$$

where t is a monotonic parameter, and s is the slope.

To generate a logarithmic spiral, the data are derived from the Cartesian coordinates

$$x = r*\cos(ang) + offset;$$

$$y = r*\sin(ang) + offset;$$

$$ang = (t*2*\pi)/360;$$

$$r = t/12.0;$$

where t is a monotonic parameter; in our implementation D.I.G.S requires offset = 120 for the starting point of the curve to be at the centre of the display device; r is the increasing radius of curvature.

An astroid can be constructed from

$$ang = (t*2*\pi)/360$$

$$ang2 = (ang*c_1)/c_2$$

$$x = c_1*\cos(ang) + c_2*\cos(ang) + offset;$$

$$y = c_1*\sin(ang) - c_2*\sin(ang) + offset;$$

The parameters c_1 and c_2 are such that $c_1 = k\,c_2$ will produce k+1 cusps, t is a monotonic parameter.

APCHAPTER5.2

A set of criteria by which the approximation algorithms can be evaluated are discussed in this appendix; and a methodology which uses these criteria to determine the 'best" data reduction algorithm is presented.

The criteria in their order of importance are:

1. **The accuracy of the representation.**
2. **The efficiency of the representation.**
3. **The compactness of the representation**

The first criterion is concerned with the amount of error introduced by the process of choosing the best representative points of the pen trajectory.

The second criterion measures two types of efficiency. The first is the efficiency of the algorithm responsible for choosing the points which matter along the pen trajectory. The second is the efficiency of algorithms that may be used to reconstruct the pen trajectory from the chosen points.

The third criterion is designed to measure the degree of data reduction achieved by the point choosing process.

The methodology for evaluating the point reduction algorithm is to consider the above criteria, to specify in more detail what is required by each criterion, to identify the relative value of each criteria, to analyze, measure and test each algorithm relative to these criteria and finally to bring together these results to make overall conclusions as to choosing the best algorithm.

APCHAPTER5.3

Throughout chapters 5, 6, 7, 8 we are concerned with a type of accuracy which we will call " visual accuracy ". Two freehand generated curves are visually acceptable, if to the average human eye and mind, they appear to be the same curve. If curves are dissimilar then a visual accuracy measure is low. Such a measure should ignore small visually insignificant errors and concentrate on regions where large discrepancies occur. The accuracy criteria is concerned with " visual accuracy ".

When trying to choose the relevant points of a pen trajectory, the following approach is taken. We firstly specify an accuracy tolerance, then execute the algorithm which finds the significant points within the representation space which has the required accuracy. Thus the accuracy criteria guides the choosing process.

In evaluating algorithms, we will often base the comparisons on the accuracy achieved in approximating a set of canonical freehand generated curves. For example, to evaluate compactness, we will compare the number of relevant points produced by the algorithm. To evaluate the efficiency criteria, we will compare the computing times.

APCHAPTER7.1

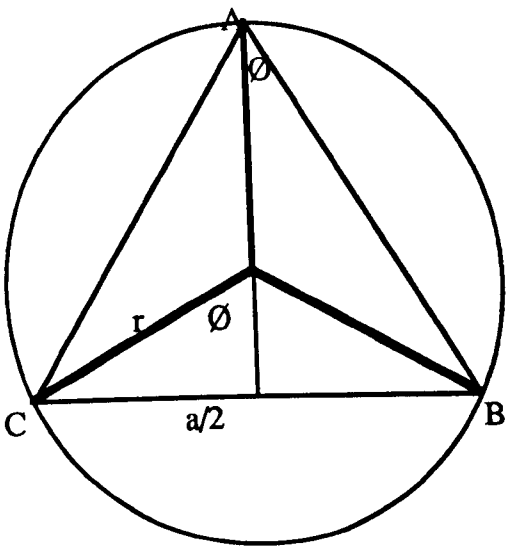


Fig.a

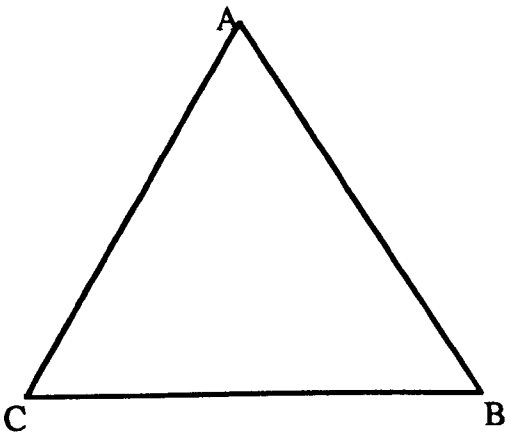


Fig.b

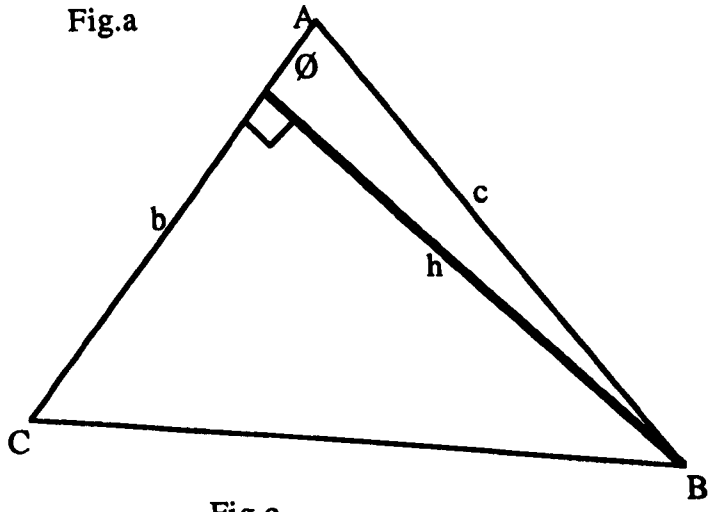


Fig.c

Using elementary results of geometry, we can see that :

From Fig.a , $\sin\emptyset = a / 2r$ (1)

Assuming that the coordinates of A, B, C are respectively (x_{t-1}, y_{t-1}) , (x_t, y_t), (x_{t+1}, y_{t+1}) then the area of the triangle ABC is

Area = $\Delta/2$ (2)

where Δ is the determinant of the 3 by 3 matrix

$$\begin{vmatrix} x_{t-1} & y_{t-1} & 1 \\ x_t & y_t & 1 \\ x_{t+1} & y_{t+1} & 1 \end{vmatrix}$$

APCHAPTER7.2

]

From Fig.c we have the results

$\sin\emptyset = h / 2$ (3)

Area = $bh / 2$ (4)

Equations (2) and (4) imply $bh = \Delta$ (5)

equations (1) and (3) imply $h / c = a / 2r$

so that $2r = ac / h$ (6)

Equations (5) and (6) imply $2r = abc / \Delta$ (7)

As a , b, c are distances, the radius r is found to be

$$r = (\sqrt{(x_{t-1} - x_t)^2 + (y_{t-1} - y_t)^2}\ \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}\ \sqrt{(x_{t+1} - x_{t-1})^2 + (y_{t+1} - y_{t-1})^2}\ )/2\Delta$$

if A, B, C are respectively represented by qt-1 , qt, qt+1

$$\mathbf{r = (|qt\text{-}1 - qt|\ |qt+1 - qt|\ |qt - qt\text{-}2|\ ) / 2\Delta}$$

BIBLIOGRAPHY

ABR63 N. Abramson : 'Information theory and coding' McGraw-Hill, New York, 1963.

AKIMA70 H. Akima : A new method of interpolation and smooth curve fitting based on local procedures, J.ACM, Vol. 17, (1970), pp 589-602.

AKIYAMA83 K. Akiyama, T. Kishomoto, Y. Sato : Simultaneous voice and handwriting signals transmission, applying voice interleaving technique. Kenkyu Jitsuyoka Hokoku (E.C.L Tech. Journal), NTT, Japan, vol.32, no 3, pp. 755-764, 1983.

BALL78 A. A. Ball : A simple specification of the parametric cubic segment, Computer Aided Design 10, 1978.

BORD78 J. Bordewijk : Teleboard, scribophone, and their relation to " coded text transmission", Electronic Text. communication, Munich Germany 12-15 June 1978.

BRADY68 P.T. Brady : A statistical analysis of on-off patterns in 16 conversations, Bell system Technical Journal, vol. 47, no 1, pp 73-91, 1968.

BRODLIE80 K.W. Brodlie : A review of methods for curve and function drawing, Proc. IMA conference on Mathematical methods in computer graphics and design, University of Leicester, september 1978. Publ. Academic Press, London, 1980.

CATT69 K.W. Cattermole : " Principles of pulse code modulation " London, ILIFFE, Books Ltd, 1969; pp 125-241.

BIBLIOGRAPY. 2

CCITT81 Incremental coding for alphageometric videotex, Question 24/VIII, Study group V/III, delayed contribution D75, October 1981.

CCITT85 Proposal on the presentation protocol for the basic telewriting terminal, Question 17/VIII, Study group V/III, contribution 16, March 1985.

CCITT85 Zone coding and incremental trace coding: an experimental comparison, Question 17/VIII, Study group V/III, delayed contribution D42, June 1985.

CCITT85 Zone coding and incremental trace coding: some computations, Question 17/VIII, Study group V/III, delayed contribution D43, June 1985.

CHOW71 M. Chow, variable length Redundancy Removal for differentially Coded Video Telephone signals, IEEE Trans. Communication Technology, COM-19 (6) pp 923-926.

COUEGN80 Philippe Coueignoux : Computer generation of colored planar patterns on tv like rasters, Proceedings of the IEEE, vol.68, no7, july 1980.

COUEGN81 Philippe Coueignoux : character generation by computer Computer graphics and image processing 16, 240-269, 1981

CRAMPIN85 M. Crampin, R. GUIFO GUIFO, G.A Read : Linear approximation of curves with bounded curvature and a data reduction algorithm, Computer Aided design, volume 17, number 6 July / august 1985.

CROCHIERE81 R. E. Crochiere : Decimators and Interpolators, Proceedings of the IEEE, vol.69, No. 3, March 1981.

DAGN78 J. P. Dagnelie : Teleboard system. Munckner Kreis. June 12-16, 1978.

DAGN79 J.P. Dagnelie Telewriting : Teleinformatics 79, IFIP Boutmy / Danthine (eds).

DANIEL82 P.E. Danielson : Polygonal approximation of digital curves. IBM technical Disclosure Bulletin, Vol. 24, No. 11B April 1982.

DANIEL83 P.E. Danielson, K. Wall : A new method for polygonal approximation of digitized curves, Proc. III Scand. Conf. on image Processing, 60-66. 1983.

DEBOOR78 Carl de Boor : A practical guide to splines, Springer, Berlin (1978)

DEL80 Delius : Simultaneous transmission of speech and handwriting over a telephone channel, NTG-Fachberichte Vol. 74 pp 254-261 30 sept -3 oct 1980, VDE, Berlin.

DES78 J. Dessimoz : Visual identification and locations in a multi-object environment by contour tracking and curvature description, Proc. 8th International symposium on industrial robots, Stuggart, Germany, pp 764 - 777, May- June 1978.

FOLEY82 J.D. Foley, A. Van Dam : Fundamentals of interactive computer graphics, Addison-Wesley, 1982, pages 514-523.

FORREST68 A. R. Forrest : Curves and surfaces for computer-aided design (PH.D. Thesis) Joint computer-aided design group, computer Laboratory, Cambridge University July 1968.

FORREST72 A.R. Forrest : Interactive interpolation and approximations by Bezier polynomials, Comput. J., 15, 71-79, 1972.

FRANCIS81 A Francis : Cyclops - Graphics for videotext, Siggraph '81.

FREEMAN74 H. Freeman : Computer processing of line drawing images, A.C.M Computing Surveys, Vol.6, no 1, Jan. 1974, pp 57-97.

FREEMAN78 H. Freeman : Application of the generalized chain coding scheme to map data processing, in Proc. IEEE Comput. Soc. Pattern recognition image processing, chicago, May 1978, pp 220-226.

GILOI78 W.K. Giloi : Interactive computer graphics (Prentice Hall, 1978)

HAM80 R.W. Hamming : Information Theory, book, 1980

HAM77 R.W. Hamming : Digital filters, (Prentice-Hall, NJ, 1977)

HUANG77 T.S Huang : 'Coding of two-tone images'; I.E.E.E Transactions, 1977, COM-25, pp 1406 - 1424.

IEEE80 Proceedings IEEE, Special issue on the encoding of graphics vol. 68, no. 7., 1980.

IEEE81 Special isssue on picture communication systems, vol. COM-29, no 12, 1981.

IEEE85 Proceedings IEEE, Special issue on visual communication systems, vol. 73, no. 2., 1985.

ISHII79 A. Ishii et al : A consideration on voice and figure combined transmission lines using a public telephone line, Papers of Technical group on Switching Engineering, IECE (Japan), SE79-45, 1979.

JEL68 F. Jelinek : Probabilistic Information theory. McGraw-Hill, New York, 1968.

KEG77 A. Kegel, and J.H Bons : On the digital processing and transmission of handwriting and sketeching, Conference Proceedings Eurocon 3-7 May 1977, Venice.

K00L80 Kool : The Scribophone, a graphic telecommunication system. Philipps, Technical Journal Jan. 1980

KAISER66 J.F Kaiser , F.F Kuo, Eds : Digital filters, " in Systems Analysis by Digital Computer. N.Y, 1966.

KAISER74 J.F Kaiser : "Non recursive digital filter design", in Proceedings 1974 IEEE Int. Synp. Circuit Syst. 1974.

KAROW87 P. Karow : Digital Formats for Typefaces, URW Verlag Hamburg 1987.

KNUTH79 D.E KNUTH : TEX and METAFONT, New directions in Typesetting, Digital press and American Mathematical Society 1979.

KITCH83 L. Kitchen and A. Rosenfeld : Gray level corner detection, Pattern recognition 4 (6), april 1983.

KUROZ82 Y. Kurozomi, W.A. Davis : Computer Graphics and Image processing, 19, 1982, pp 248-264.

LANG82 D.J. Langridge : Curve encoding and the detection of discontinuities, Computer vision, Graphics, Image Process 20, 1982, 58-71.

LORIG81 B. Lorig, J.C, Rahuel : Combined telewriting - videotext terminal, l'echo des recherches, english issue 1981, pp 27 -34.

LORIG80 B.LORIG, C. ROUX, J.C. RAHUEL : Teleboard system Computer Graphics 80.

MANN72 J.R Manning : Continuity conditions for splines curves, Shoe and Allied trades Research association, Satra House, Rockingham Road, Kettering, Northamptonshire (1972)

MAKO80 Y. Makoto and Y. Yasumoto : Adtaptive linear predictive coding of handwriting signals, Signal Processing

Theories and Applications, E.U.R.A.S.I.P, 1980, pp 503-507.

MCLAUGLIN83 H.W. Mclaughlin : Shape preserving planar interpolation; An algorithm, IEEE CG &A, May / june 1983 pp. 58-67

MED65 J.E. Medlin : Sampled-data Prediction for telemetry Bandwidth compression" IEEE transactions On Space Electronics and Telemetry, March 1965, p.29.

MIDGEL79 J.E Midgeley : Isotropic four-point interpolation, computer Graphics and image processing, 1979, 9 pp 192 - 196

MYL72 G. Myles, C.Fischer : "Telewriting system", U.S patent 3,706850, Dec 19, 1972.

NCC82 The National Computing Centre : Handbook of data communications. NCC Publications, The National Computing Centre Limited, Oxford Road, Manchester M1 7ED, England, 1982.

NET80 Arun, N. Netravali, John Lim : " Picture coding, a review" Proceedings of the I.E.E.E, vol.68, no 3, March 1980, pp 367-402.

NIEW73 L.R. Niewkerk : A writing tablet for converting handwriting into electrical signals, Tijdschrift van het N.E.R.G, vol 38 no 6 1973.

OPENU75 The Open University press : Noise in instrumentation systems, T291, 11, 12 and 13, 1975.

PAVL82 T. Pavlidis : Algorithms for Graphics and image processing. Computer Science press, Rockville, Md.1982

PAVL74 T. Pavlidis : Segmentation of plane curves IEEE transactions on Computers C-23, 866-870, August 1974.

PATENT73 United States Patent, No 3,732,557, May 8 1973.

PERD81 An introduction to the Matrox tv crt controller modules, Applications notes, Perdix Components Ltd, 98 Crofton Park, London, S.E.4.

PRY70 I.W. Pryke : A review of Data compression Techniques of the Redundancy Reduction Type, and associated procedures for possible augmentation by an On Board Computer. ESTEC-ESRO, Ref. No. TS/2325/IP/MA, October 1970.

RAH80 C. Rahuel et al : Graphics tablet particularly for telewriting system, U.S patent 4,225750, sep, 30, 1980.

READ77. G.A.Read : application of microelectronics to education at a distance, Educational Broadcasting International, June 1977

READ81. G.A. Read : Cyclops, Open University Press 1981.

READ83. G.A. Read : private communication.

RENNER82 G. Renner : A method of shape description for mechanical engineering practice, Computers in Industry 3, 1982 pp 137-142

RIESEN73 R. F. Riesenfeld : " Application of B-spline Approximation to Geometric Problems of Computer-aided Design," Univ. Utah Comput. Sci. Dept. UTECH-Csc-73-126, March 1973.

RITCH78 D. M. Ritchie, W.kernighan : The C programming language Prentice-Hall, Inc, 1978

ROBERGE85 J. Roberge : A new data reduction algorithm, Computer vision and image processing 1985

ROG73 J.L. Rogers : Apparatus for converting the position of a manually operated instrument into an electrical signal, U.S patent 3,767858, oct. 1973

ROG74 J.L. Rogers : Method and apparatus for conveying graphic information over a telephone quality link, U.S patent 3,851097, Nov 26 1974.

ROGER76 D.F. Rogers : Mathematical elements for computer graphics,

Mcgraw-Hill, New York, 1976

ROSEN73 A. Rosenfeld : Corner detection, IEEE transactions on Computer graphics, 1973.

ROSEN76 A, Rosenfeld, and A.C Kak : Digital picture processing, New York , Academic Press, 1976.

RUTK79 W.S. Rutkowski : shape completion, computer graphics and image processing 9, 1979, 89-101

RIST73 M.P. Ristenbatt : " Alternatives in digital communication," Proceedings IEEE, vol. 61, pp 703-721, june 1973.

SCHAFER73 R.W. Schafer : A digital Signal processing approach to interpolation, Proceedings of the IEEE, vol 61, 1973

SHAH84 A. Shah, Mubarak, and Ramesh : Detecting time varying corners, Comput. Vision, Graphics and image Process 28 1984, pp 345-355

SHAN48 C. Shannon : Mathematical theory of communication, Bell technical Journal, 1948.

SKLAN80 J. Sklansky and V. Gonzalez : Fast polygonal approximation of digitized curves, PRIP Proc79-80, pp 604 - 609

SMOL76 G. Smol, M. Hamer, M. Hills : Telecommunications A system approach, London GEORGE ALLEN & UNWIN LTD, 1976, pp 147-259.

SMOL81 G. Smol : "PDMAKE, VAXHL, RECEIVE, DATACONV" a set of application programs for recording handwriting data bidirectional link between the North Star Horizon microcomputer and the minicomputer VAX11/750, and data conversion program for structure management; Departement of Electronics and Telecommunication, Open University, Milton keynes, England, 1981.

SUM79 Summagraphics : Bit Pad one, User manual by Summagraphics Corporation 35 Brentwood Ave, Fairfiled, Connecticut 06430, Sept 1979.

TANEN81 A.S.Tanenbaum : Computer networks Prentice / Hall International editions 1981.

TOMIO83 Tomio, Kishimoto, Yuichi Sato : Zone coding, a new coding technique for telewriting signals, I.C.C , I.E.E.E 1984 pp 975 - 979.

WEBER65 D.R. Weber : " A synopsis on Data Compression", Proceedings of the 1965, National Telemetring conference (NTC) pp. 9-16, A 65-2419.

WITTEN87 Ian Witten : Radford M. Neal, and J. G. Cleary Arithmetic coding for data compression, Communications of the ACM June 1987, Volume 30, Number 6, pp 520-540

YAMA78 F. Yamaguchi : A new curve fitting method using a CRT computer display. Computer graphic and image processing, 7 (1978 425-437)

YUI82 Yuichi, Sato and Taichi, Nakamura : Transactions of the I.E.C.E of Japan section B (in Japanese), predictive encoding method for handwriting signals, 1982, vol, J65, B, pt 2.